Building Systems Resilient to Failures, Partitions and Slowdowns

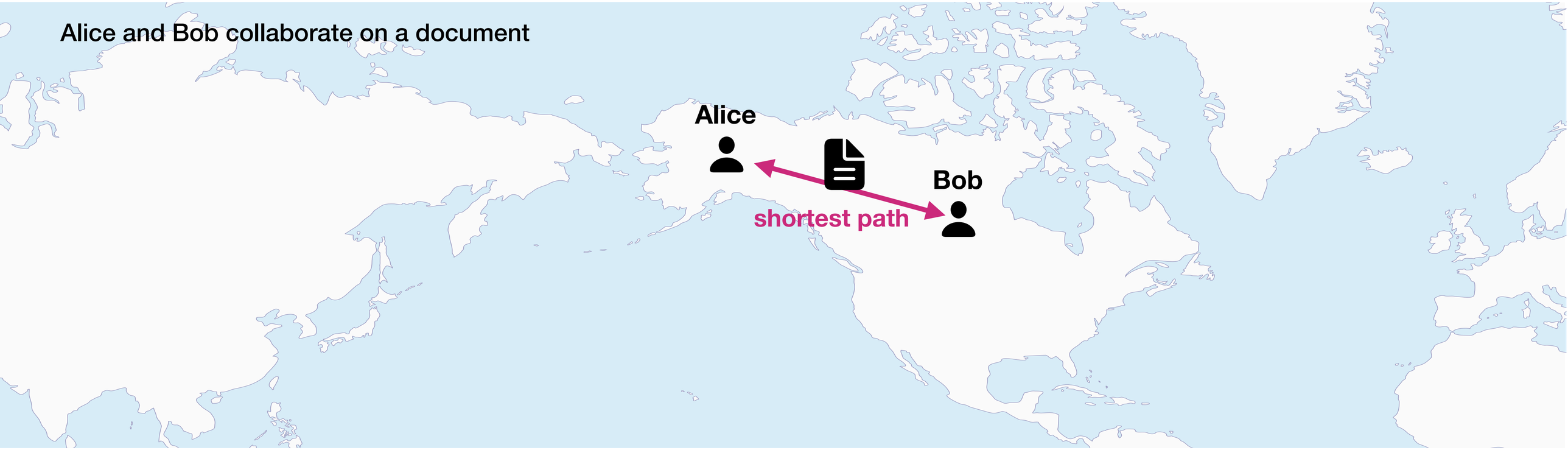
PhD Thesis Defense
30.11.2022

Cristina Băsescu

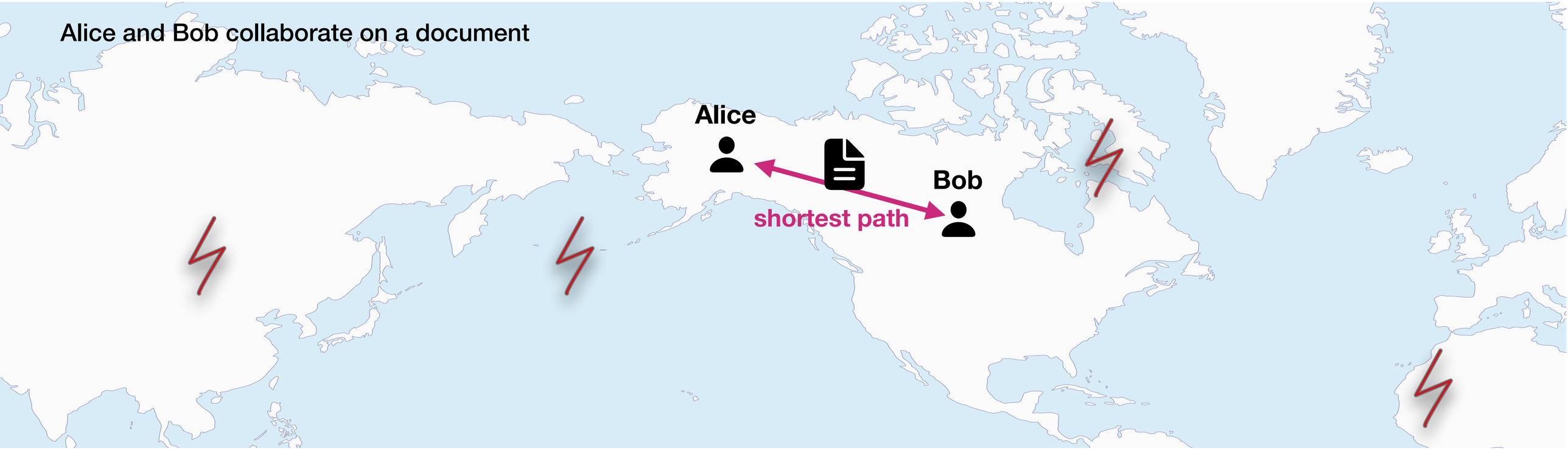
EPFL

An ideal situation

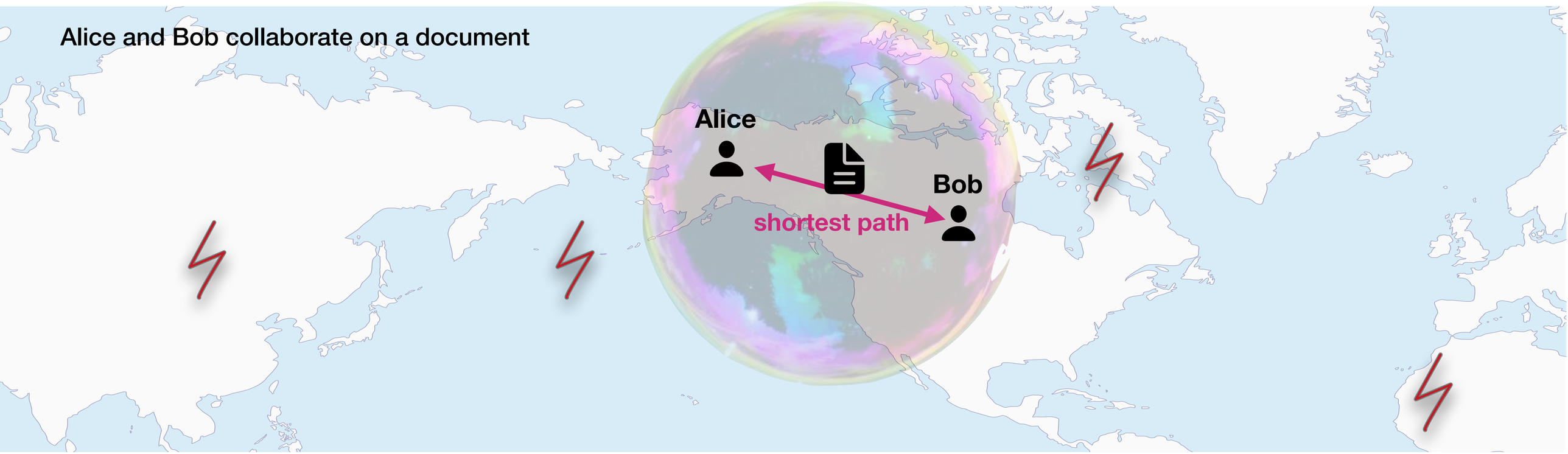
An ideal situation



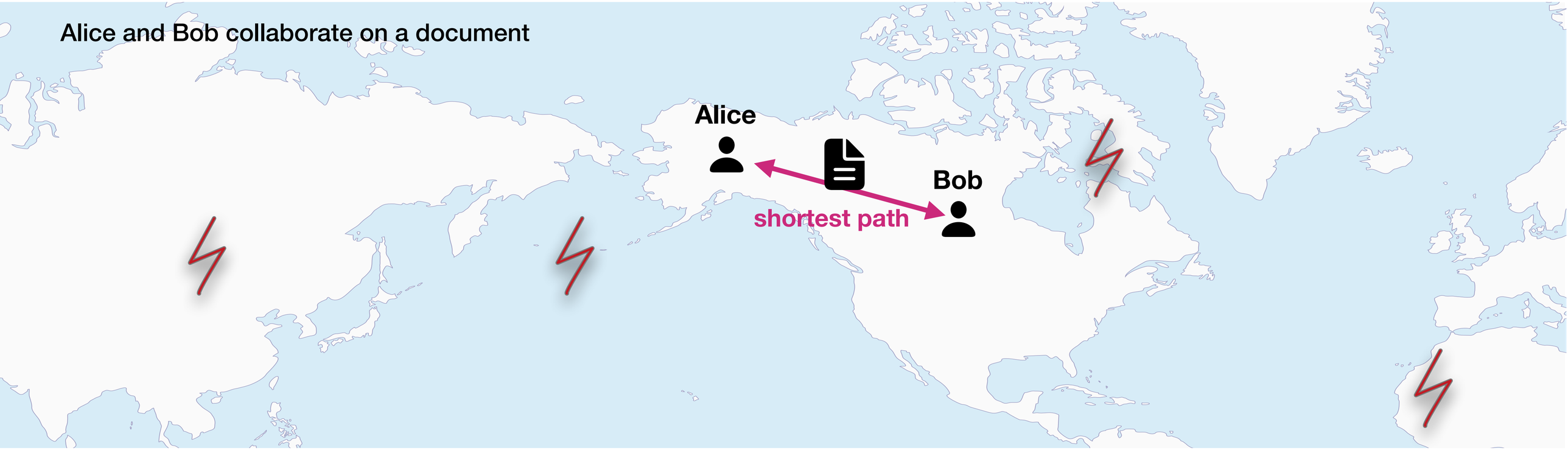
An ideal situation



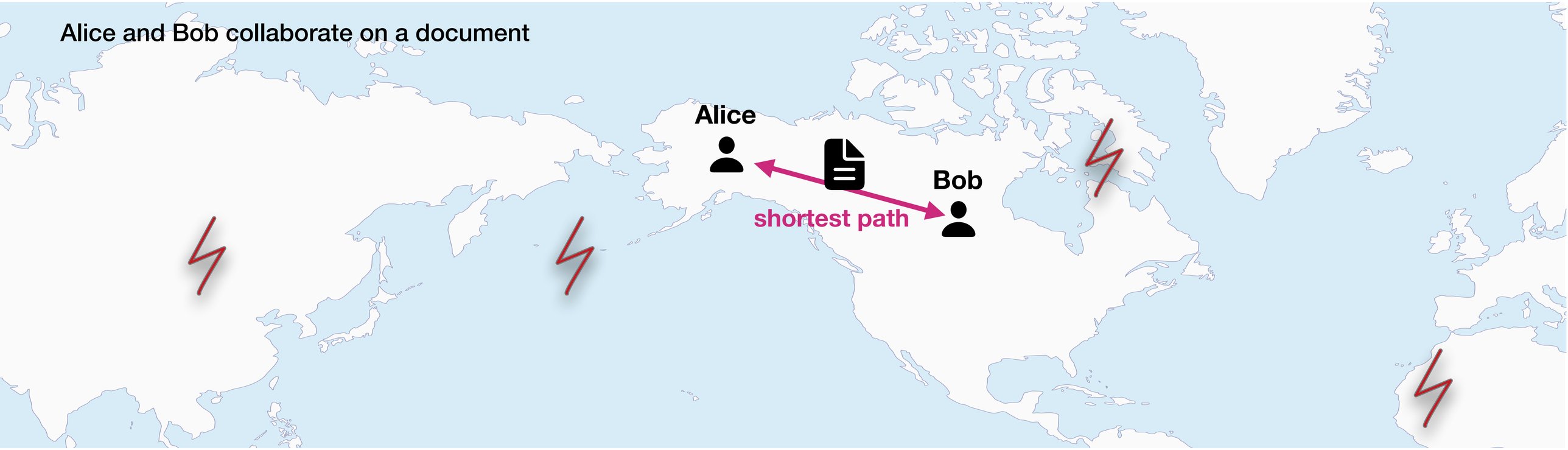
An ideal situation



An ideal situation

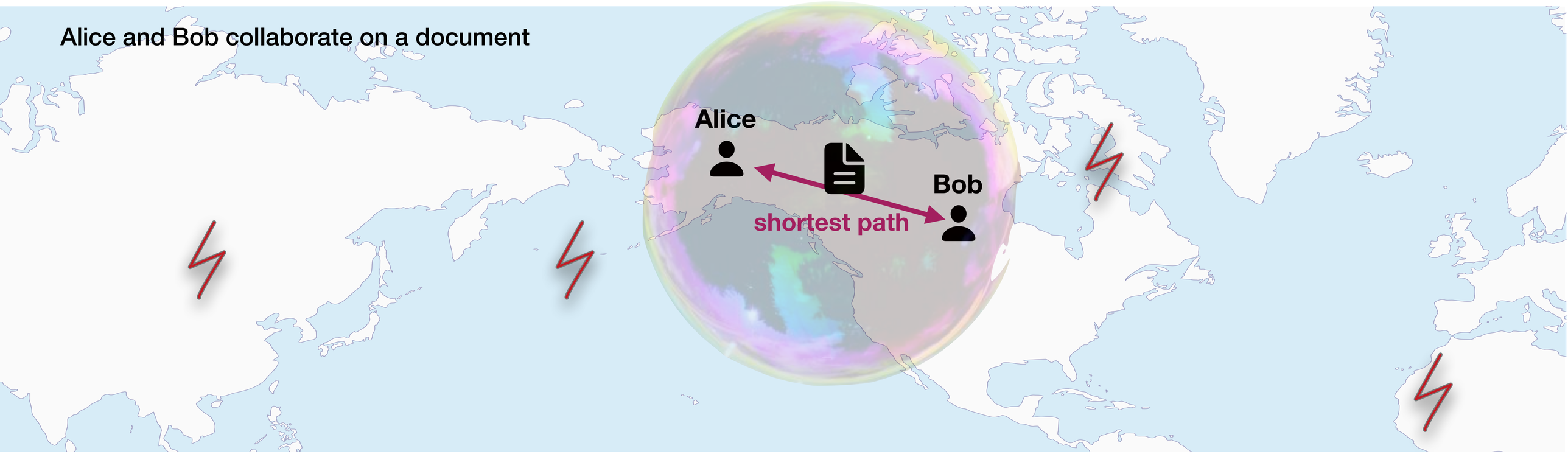


An ideal situation



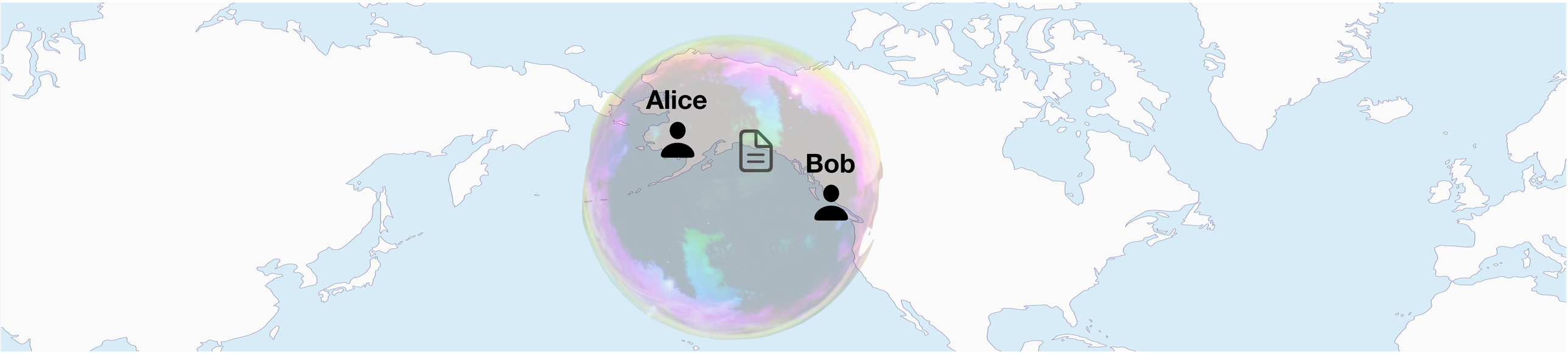
Goal: resilience

An ideal situation

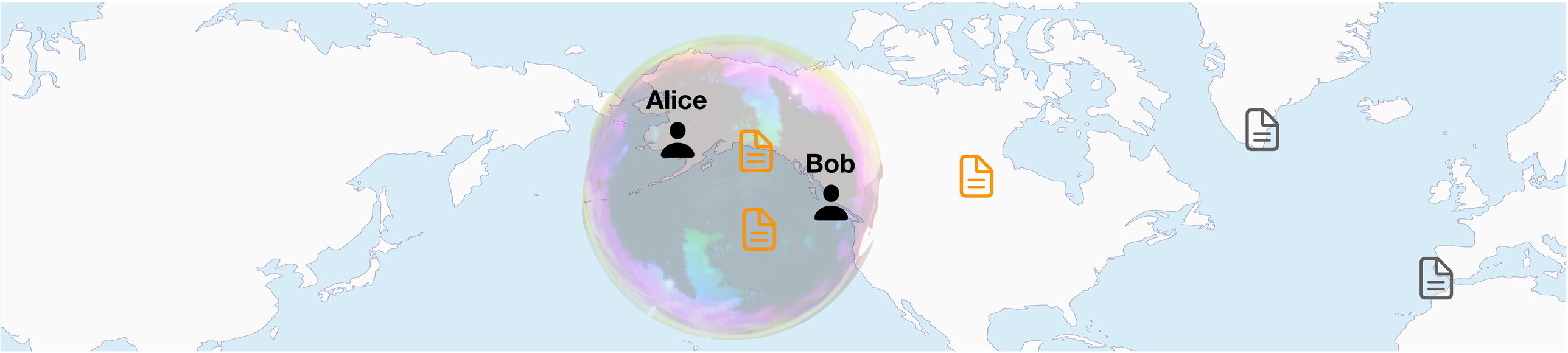


Goal: resilience

A more realistic view

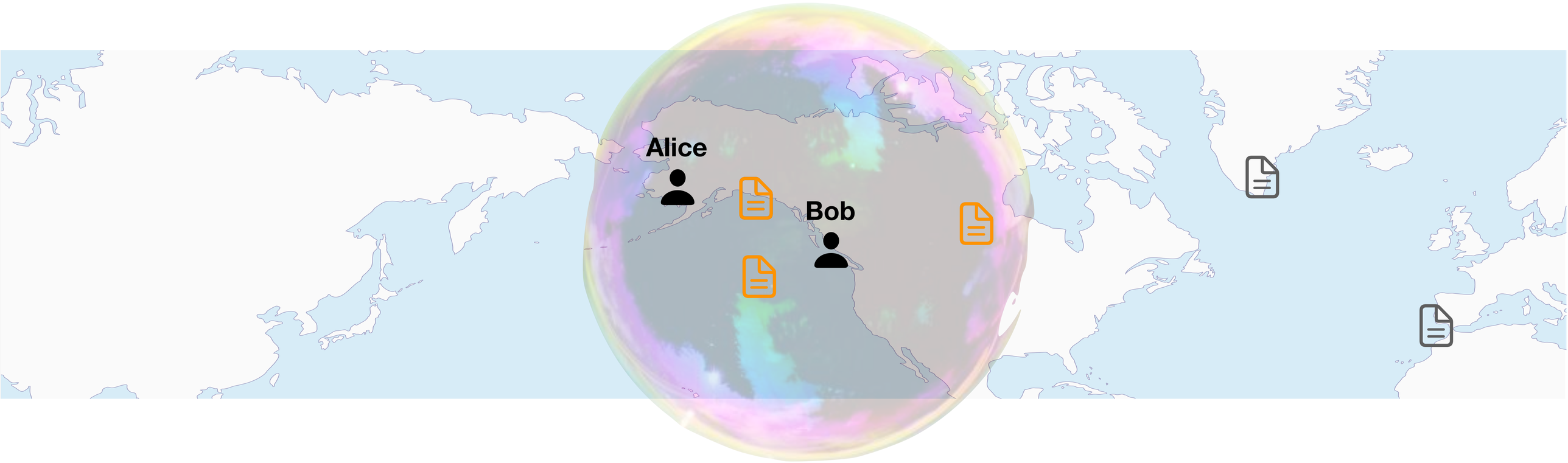


A more realistic view

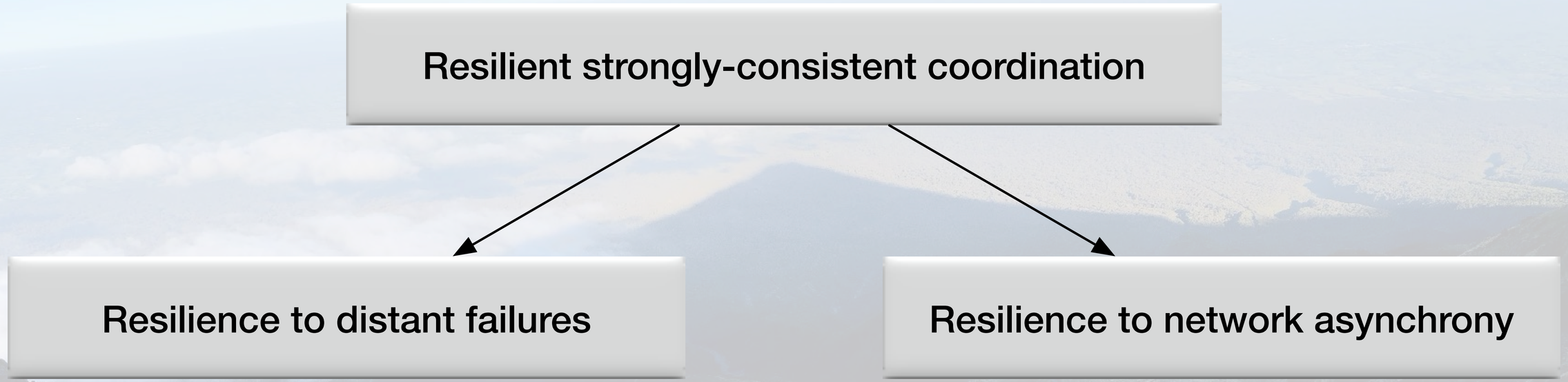


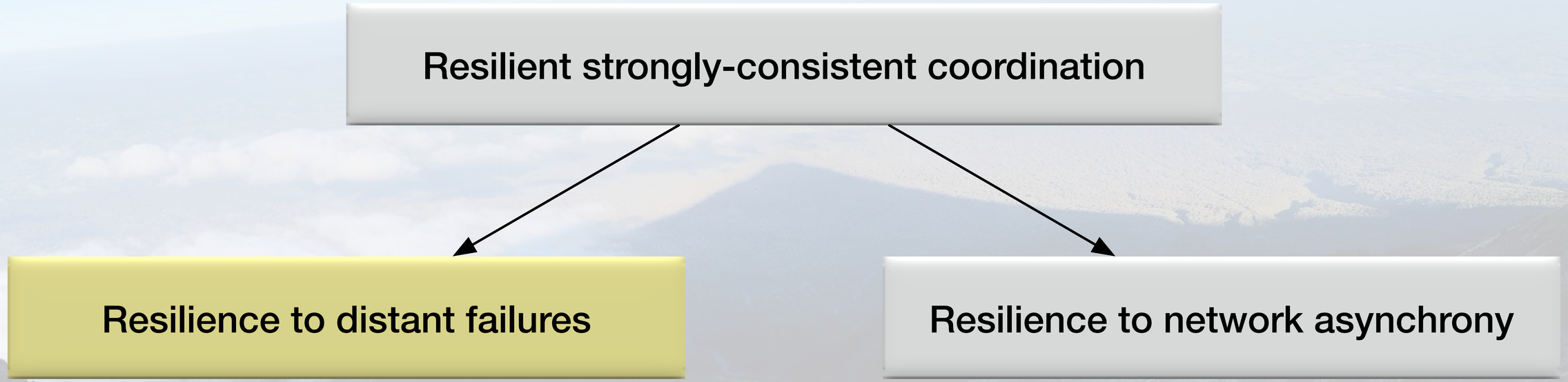
Replication masks **individual replica failures**, but adds **dependency on quorum**

A more realistic view

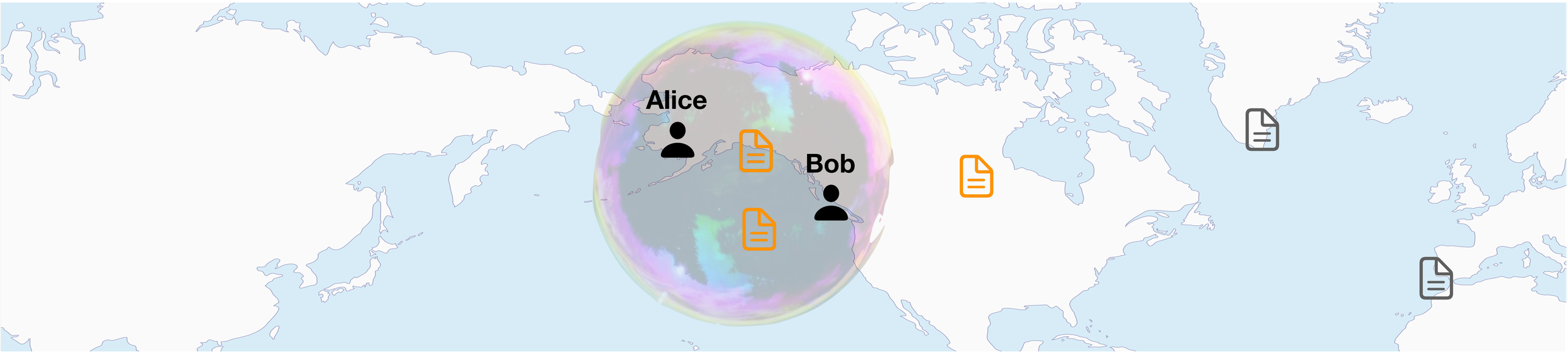


Replication masks **individual replica failures**, but adds **dependency on quorum**

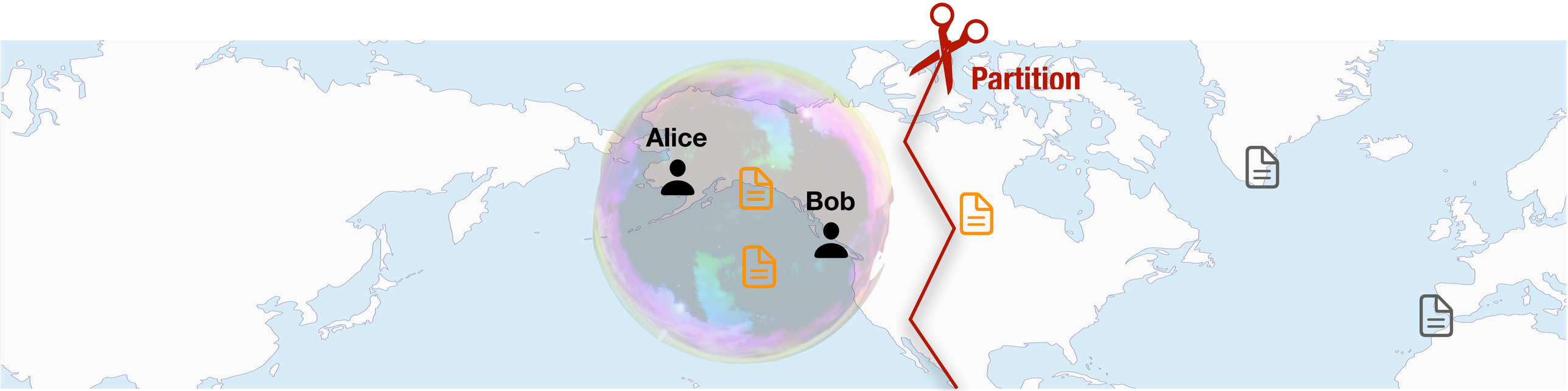

Resilient strongly-consistent coordination
Resilience to distant failures
Resilience to network asynchrony


Resilient strongly-consistent coordination
Resilience to distant failures
Resilience to network asynchrony

Challenge: Local resilience in a global world



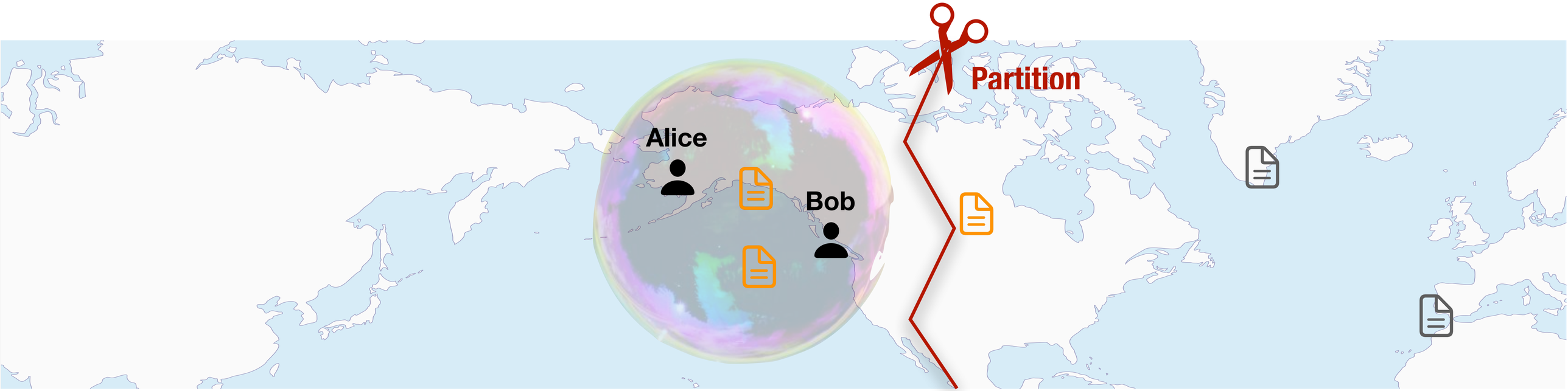
Challenge: Local resilience in a global world



CAP theorem: **pick two** of strong consistency, availability, partition tolerance

* Seth Gilbert and Nancy Lynch. Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. *ACM SIGACT News*, 33(2), 2002.

Challenge: Local resilience in a global world



Rare events?
Naïve simplification

CAP theorem: **pick two** of strong consistency, availability, partition tolerance

* Seth Gilbert and Nancy Lynch. Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. *ACM SIGACT News*, 33(2), 2002.

Gray failures

Individual replica failure

Full partitions

Gray failures

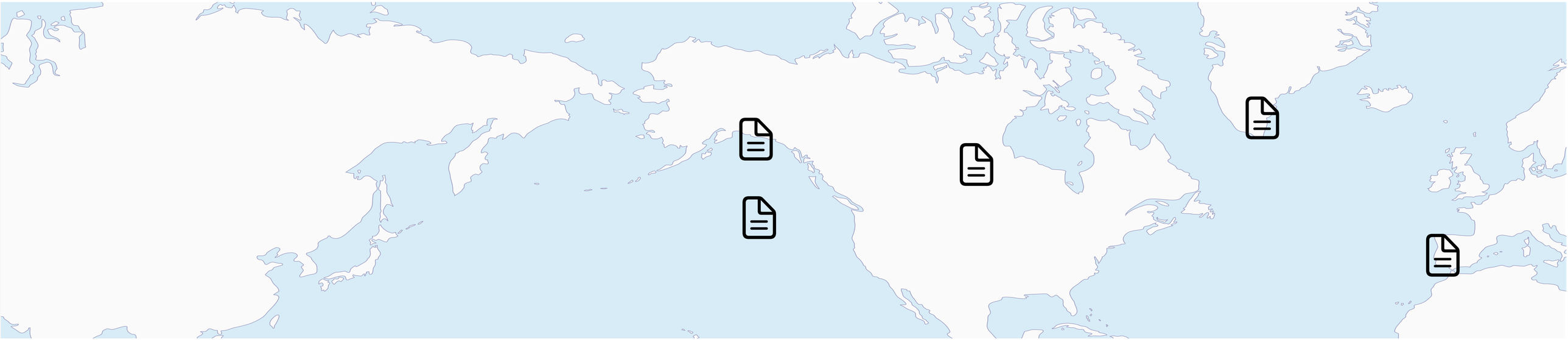
Individual replica failure

Full partitions

Gray failures

Individual replica failure

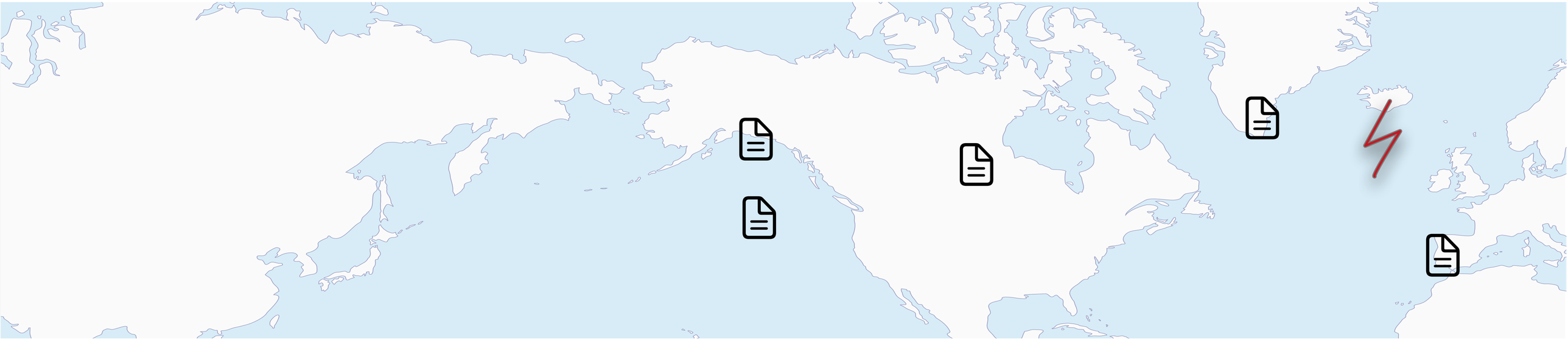
Full partitions



Gray failures

Individual replica failure

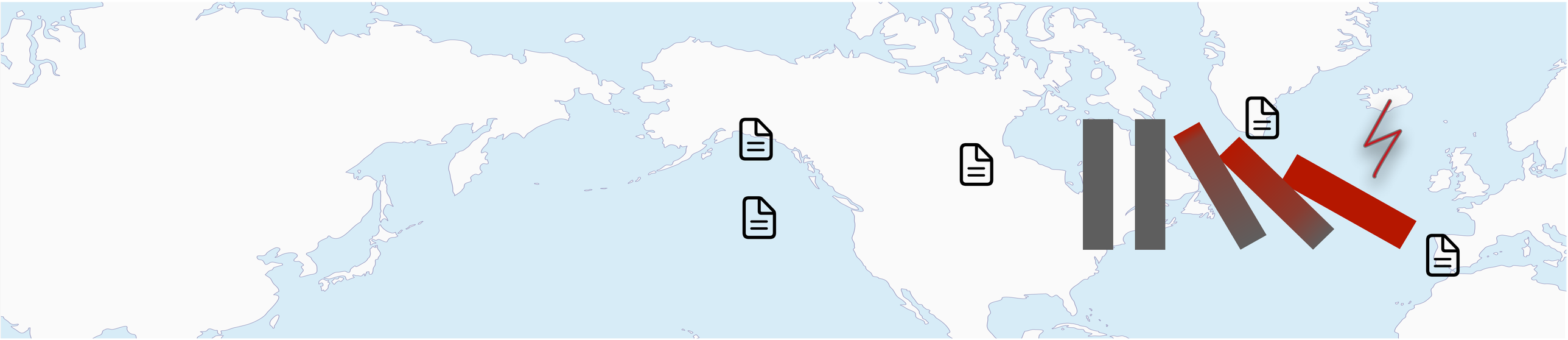
Full partitions



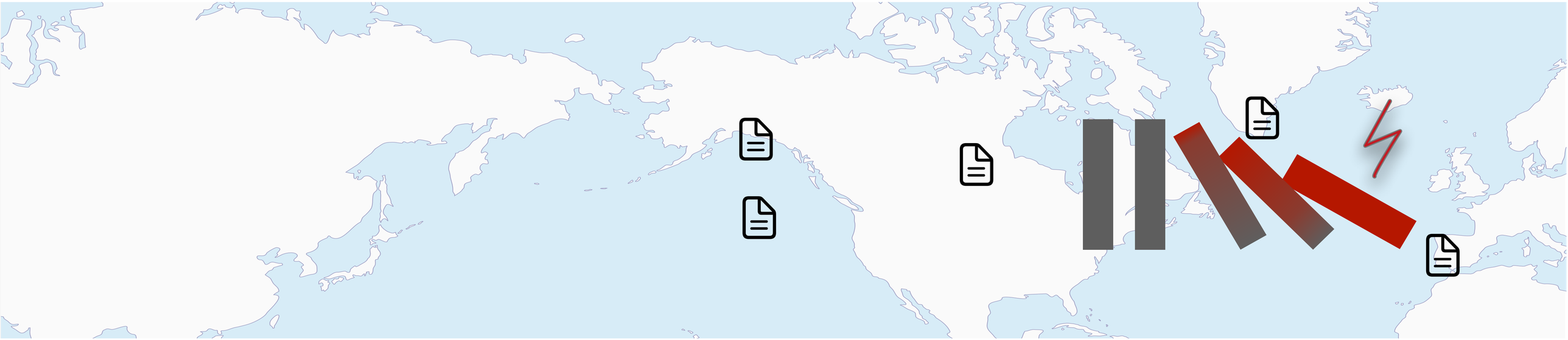
Gray failures

Individual replica failure

Full partitions



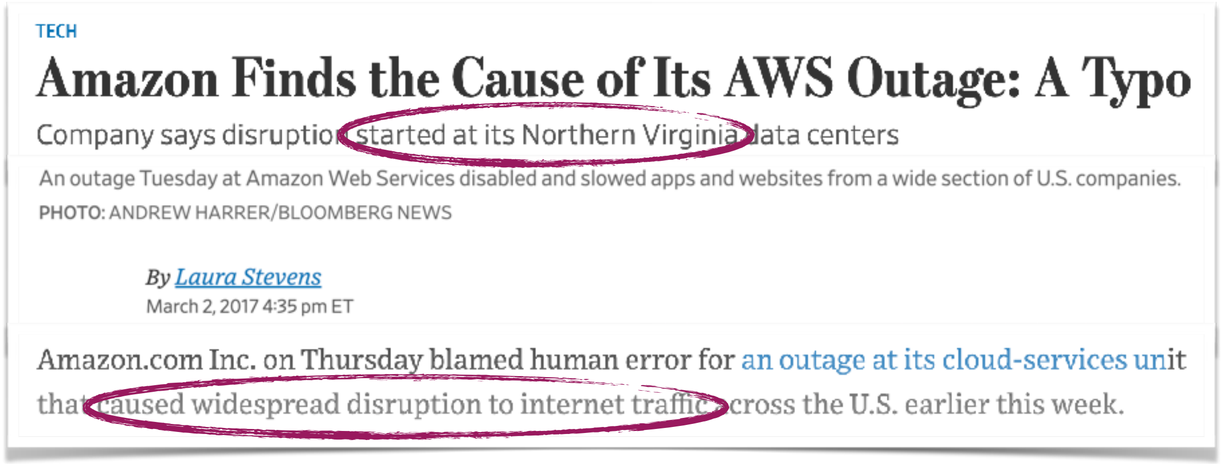
Gray failures



Resilience ⇔ withstanding all failure types and combinations

Why is it naïve to simplify partitions

Why is it naïve to simplify partitions



TECH

Amazon Finds the Cause of Its AWS Outage: A Typo

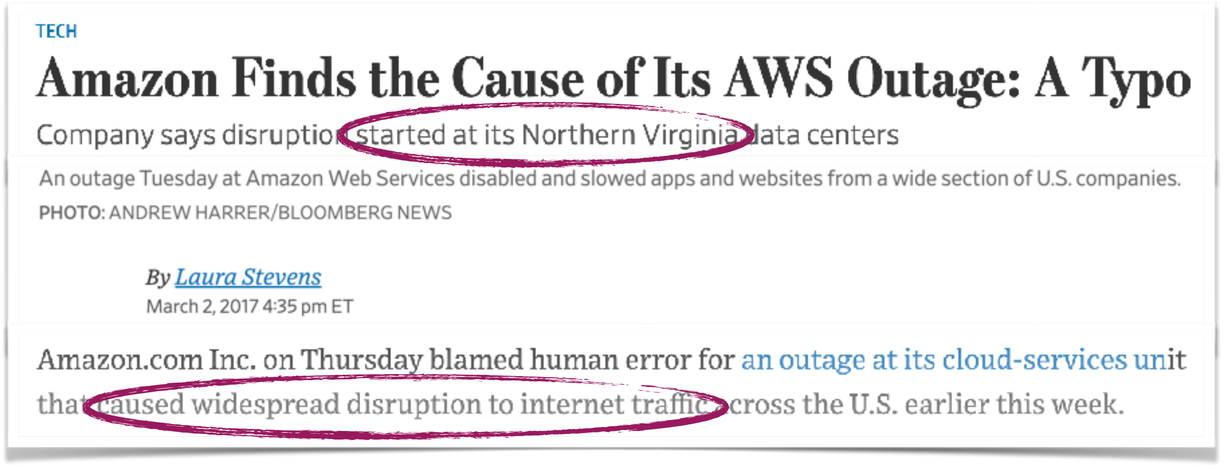
Company says disruption started at its Northern Virginia data centers

An outage Tuesday at Amazon Web Services disabled and slowed apps and websites from a wide section of U.S. companies.
PHOTO: ANDREW HARRER/BLOOMBERG NEWS

By Laura Stevens
March 2, 2017 4:35 pm ET

Amazon.com Inc. on Thursday blamed human error for an outage at its cloud-services unit that caused widespread disruption to internet traffic across the U.S. earlier this week.

Why is it naïve to simplify partitions



TECH

Amazon Finds the Cause of Its AWS Outage: A Typo

Company says disruption started at its Northern Virginia data centers

An outage Tuesday at Amazon Web Services disabled and slowed apps and websites from a wide section of U.S. companies.
PHOTO: ANDREW HARRER/BLOOMBERG NEWS

By Laura Stevens
March 2, 2017 4:35 pm ET

Amazon.com Inc. on Thursday blamed human error for an outage at its cloud-services unit that caused widespread disruption to internet traffic across the U.S. earlier this week.

Fastly says single customer triggered bug behind mass internet outage

Flaw was introduced in May and lay dormant until a customer updated their settings, firm says

A Byzantine failure in the real world

27/11/2020

Tom Lianza Chris Snook

An analysis of the Cloudflare API availability incident on 2020-11-02

When we review design documents at Cloudflare, we are always on the lookout for Single Points of Failure (SPOFs). Eliminating these is a necessary step in architecting a system you can be confident in. Ironically, when you're designing a system with built-in redundancy, you spend most of your time thinking about how well it functions when that redundancy is lost.

On November 2, 2020, Cloudflare had an incident that impacted the availability of the API and dashboard for six hours and 33 minutes. During this incident, the success rate

Why is it naïve to simplify partitions

TECH

Amazon Finds the Cause of Its AWS Outage: A Typo

Company says disruption started at its Northern Virginia data centers

An outage Tuesday at Amazon Web Services disabled and slowed apps and websites from a wide section of U.S. companies.
PHOTO: ANDREW HARRER/BLOOMBERG NEWS

By Laura Stevens
March 2, 2017 4:35 pm ET

Amazon.com Inc. on Thursday blamed human error for an outage at its cloud-services unit that caused widespread disruption to internet traffic across the U.S. earlier this week.

Fastly says single customer triggered bug behind mass internet outage

Flaw was introduced in May and lay dormant until a customer updated their settings, firm says

A Byzantine failure in the real world

27/11/2020

Tom Lianza Chris Snook

An analysis of the Cloudflare API availability incident on 2020-11-02

When we review design documents at Cloudflare, we are always on the lookout for Single Points of Failure (SPOFs). Eliminating these is a necessary step in architecting a system you can be confident in. Ironically, when you're designing a system with built-in redundancy, you spend most of your time thinking about how well it functions when that redundancy is lost.

On November 2, 2020, Cloudflare had an incident that impacted the availability of the API and dashboard for six hours and 33 minutes. During this incident, the success rate

An Analysis of Network-Partitioning Failures in Cloud Systems

Ahmed Alquraan, Hatem Takruri, Mohammed Alfatafta, Samer Al-Kiswany
University of Waterloo, Canada

OSDI '18

Gray Failure: The Achilles' Heel of Cloud-Scale Systems

Peng Huang
Microsoft Research
Johns Hopkins University

Chuanxiong Guo
Microsoft Research

Lidong Zhou
Microsoft Research

Jacob R. Lorch
Microsoft Research

Yingnong Dang
Microsoft Azure

Murali Chintalapati
Microsoft Azure

Randolph Yao
Microsoft Azure

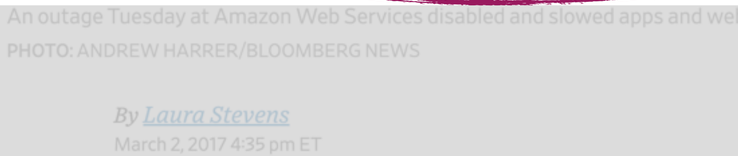
HotOS '17

Why is it naïve to simplify partitions

TECH

Amazon Finds the Cause of Its AWS Outage: A Typo

Company says disruption started at its Northern Virginia data centers



An outage Tuesday at Amazon Web Services disabled and slowed apps and websites from a wide section of U.S. companies.
PHOTO: ANDREW HARRER/BLOOMBERG NEWS

By Laura Stevens
March 2, 2017 4:35 pm ET

Amazon.com Inc. on Thursday blamed human error for an outage at its cloud-services unit that caused widespread disruption to internet traffic across the U.S. earlier this week.

Fastly says single customer triggered bug behind mass internet outage

Flaw was introduced in May and lay dormant until a customer updated their settings, firm says

A Byzantine failure in the real world

Tom Lianza Chris Snook

An analysis of the Cloudflare API availability incident on 2020-11-02

When we review design documents at Cloudflare, we are always on the lookout for Single Points of Failure (SPOFs). Eliminating these is a necessary step in architecting a system you can be confident in. Ironically, when you're designing a system with built-in redundancy, you spend most of your time thinking about how well it functions when that redundancy is lost.

On November 2, 2020, Cloudflare had an incident that impacted the availability of the API and dashboard for six hours and 33 minutes. During this incident, the success rate

Many real-world correlated failures in clouds

An Analysis of Network-Partitioning Failures in Cloud Systems

Ahmed Alquraan, Hatem Takruri, Mohammed Alfatafta, Samer Al-Kiswany
University of Waterloo, Canada

OSDI '18

Gray Failure: The Achilles' Heel of Cloud-Scale Systems

Peng Huang
Microsoft Research
Johns Hopkins University

Chuanxiong Guo
Microsoft Research

Lidong Zhou
Microsoft Research

Jacob R. Lorch
Microsoft Research

Yingnong Dang
Microsoft Azure

Murali Chintalapati
Microsoft Azure

Randolph Yao
Microsoft Azure

HotOS '17

Real-world correlated failures in blockchains

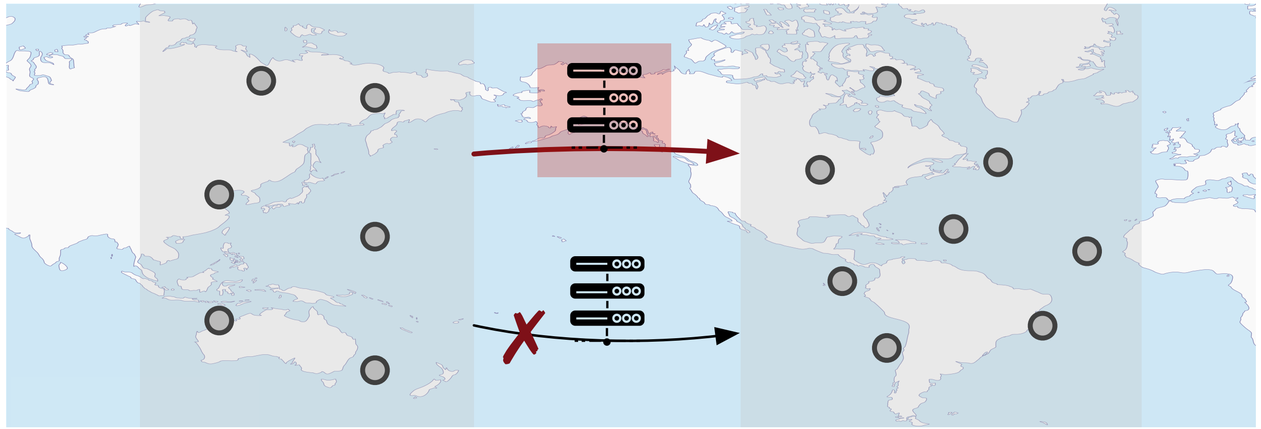
- **Blockchains are vulnerable to attacks**

Real-world correlated failures in blockchains

- **Blockchains are vulnerable to attacks**
  - Network routing attacks ⇒ unavailability or forks

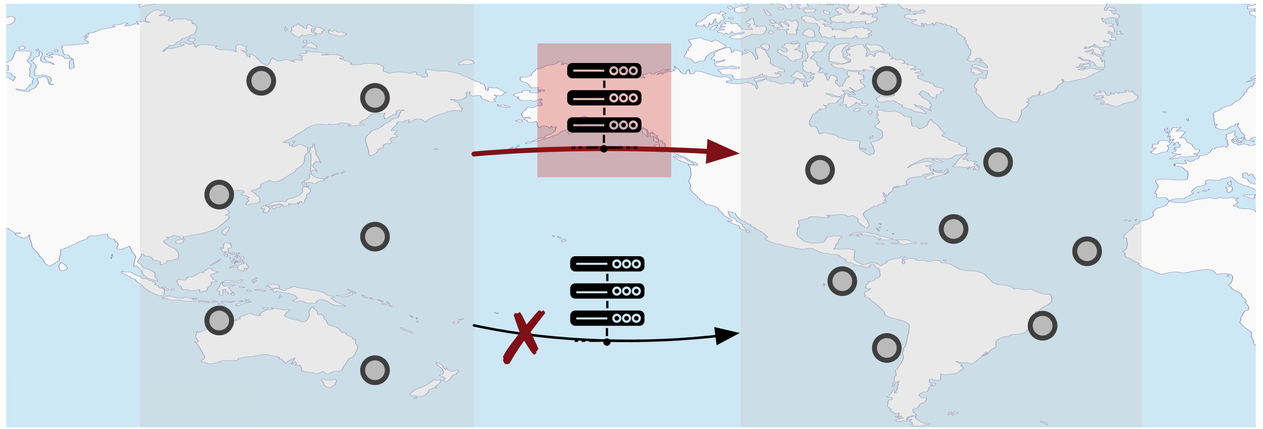
Real-world correlated failures in blockchains

- **Blockchains are vulnerable to attacks**
  - Network routing attacks ⇒ unavailability or forks



Real-world correlated failures in blockchains

- **Blockchains are vulnerable to attacks**
  - Network routing attacks ⇒ unavailability or forks



Hijacking Bitcoin: Routing Attacks on Cryptocurrencies

https://btc-hijack.ethz.ch

Maria Apostolaki
ETH Zürich
apmaria@ethz.ch

Aviv Zohar
The Hebrew University
avivz@cs.huji.ac.il

Laurent Vanbever
ETH Zürich
lvanbever@ethz.ch

Application-layer and network-layer defenses are critical for fortifying routing attacks.

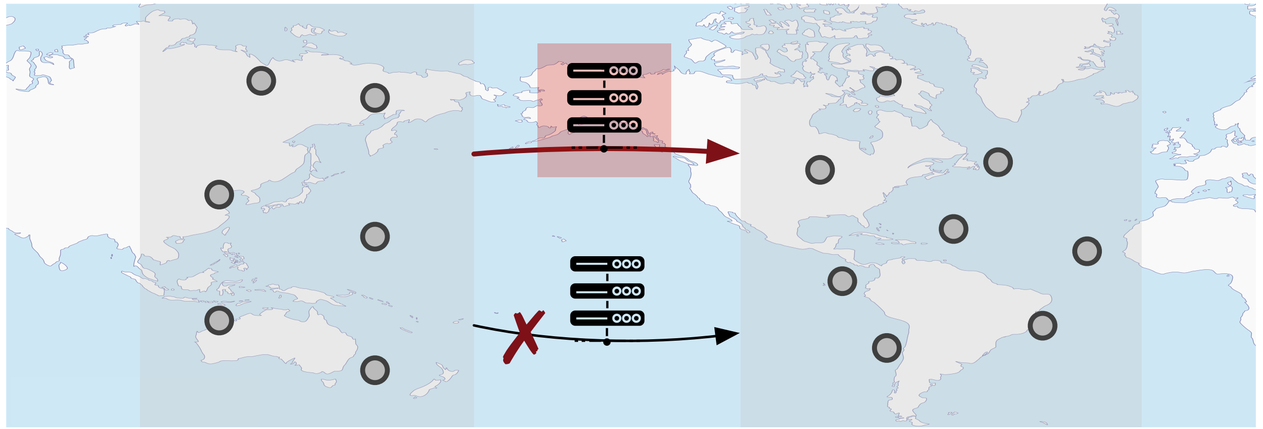
BY YIKIN SUN, MARIA APOSTOLAKI, HENRY BIRGE-LEE, LAURENT VANBEVER, JENNIFER REXFORD, MUNG CHIANG, AND PRATEEK MITTAL

Securing Internet Applications from Routing Attacks

Routing attacks on consensus. Network-level adversaries can perform routing attacks on bitcoin to partition the set of nodes into two (or more) disjoint components.[2] Consequently, the attacks disrupt the ability of the entire network to reach consensus. The adversary must divert and cut *all* the connections connecting the various components together. To do so, the adversary can perform an interception attack by hijacking the IP prefixes of each component and selectively dropping the

Real-world correlated failures in blockchains

- **Blockchains are vulnerable to attacks**
  - Network routing attacks ⇒ unavailability or forks



Hijacking Bitcoin: Routing Attacks on Cryptocurrencies

https://btc-hijack.ethz.ch

Maria Apostolaki
ETH Zürich
apmaria@ethz.ch

Aviv Zohar
The Hebrew University
avivz@cs.huji.ac.il

Laurent Vanbever
ETH Zürich
lvanbever@ethz.ch

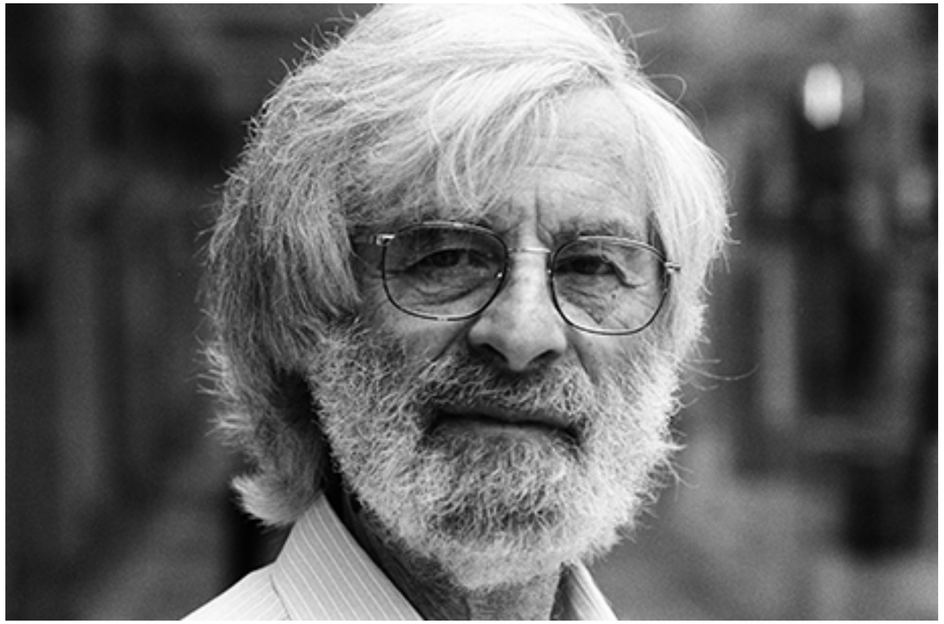
Application-layer and network-layer defenses are critical for fortifying routing attacks.

BY YIKIN SUN, MARIA APOSTOLAKI, HENRY BIRGE-LEE, LAURENT VANBEVER, JENNIFER REXFORD, MUNG CHIANG, AND PRATEEK MITTAL

Securing Internet Applications from Routing Attacks

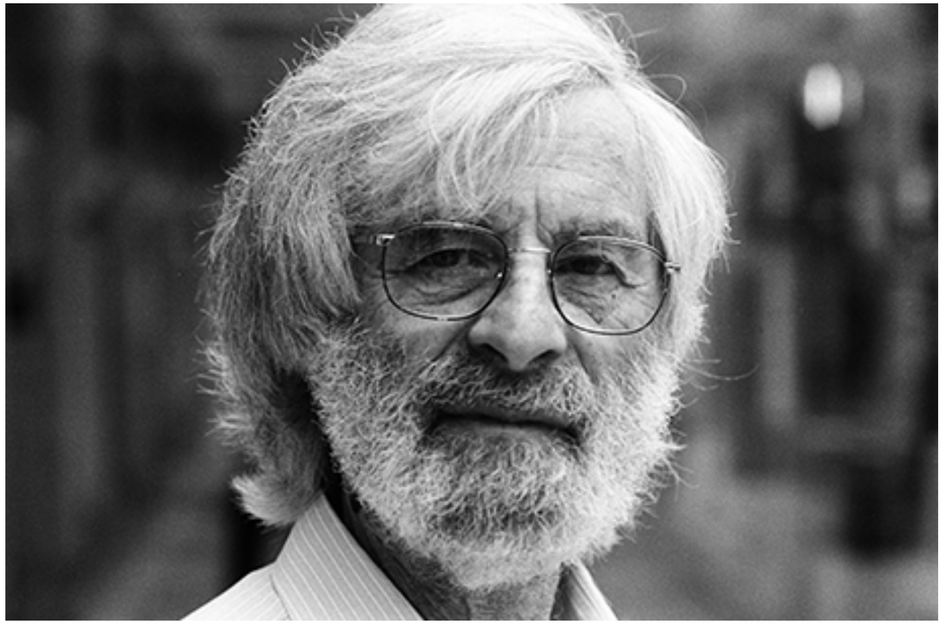
Routing attacks on consensus. Network-level adversaries can perform routing attacks on bitcoin to partition the set of nodes into two (or more) disjoint components.[2] Consequently, the attacks disrupt the ability of the entire network to reach consensus. The adversary must divert and cut *all* the connections connecting the various components together. To do so, the adversary can perform an interception attack by hijacking the IP prefixes of each component and selectively dropping the

  - Censorship / validators offline ⇒ unavailability



“A distributed system is
one in which the failure of a computer
you didn't even know existed
can render your own computer
unusable.”

(Leslie Lamport)



“A distributed system is
one in which the failure of a computer
you didn't even know existed
can render your own computer
unusable.”

(Leslie Lamport)

Can we protect Alice’s access from distant failures, partitions and slowdowns?

Insight 1

Alice cares that the service is
available for her access, not globally

Insight 1

Alice cares that the service is
available for her access, not globally

Limit exposure:
local accesses have no global dependencies

Resilient strongly-consistent coordination

Resilience to distant failures

Resilience to network asynchrony

Limit exposure

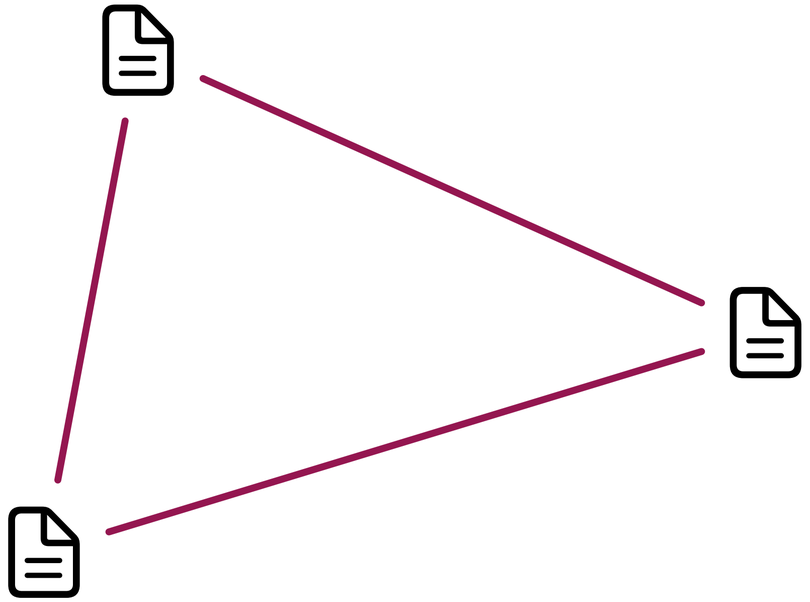
Resilient strongly-consistent coordination

Resilience to distant failures

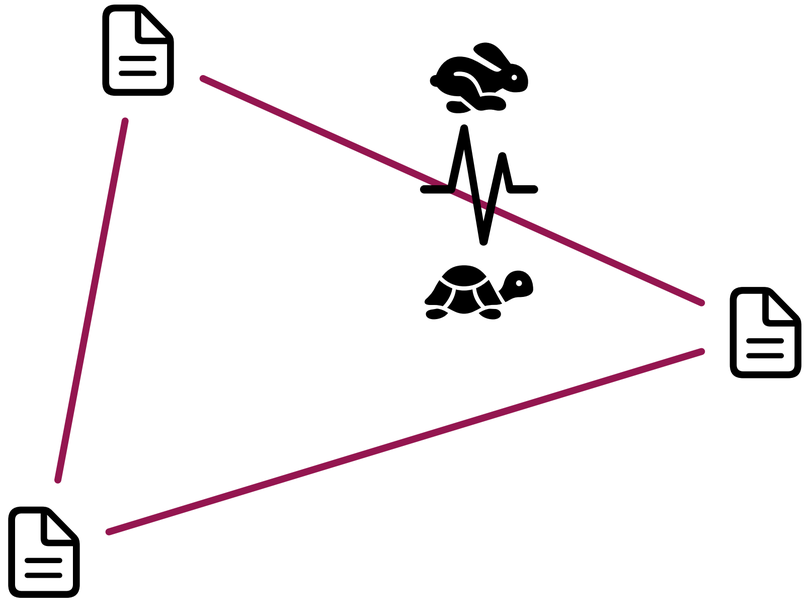
Resilience to network asynchrony

Limit exposure

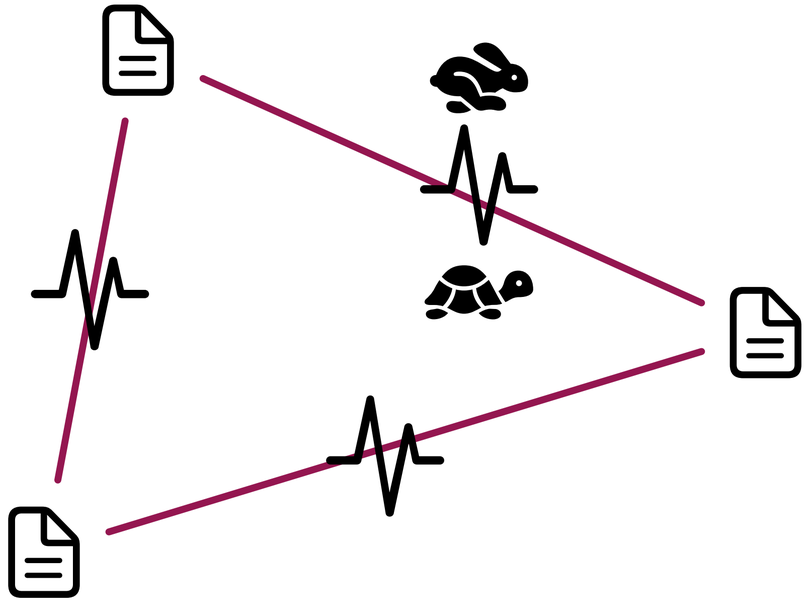
Challenge 2: Resilience despite asynchrony



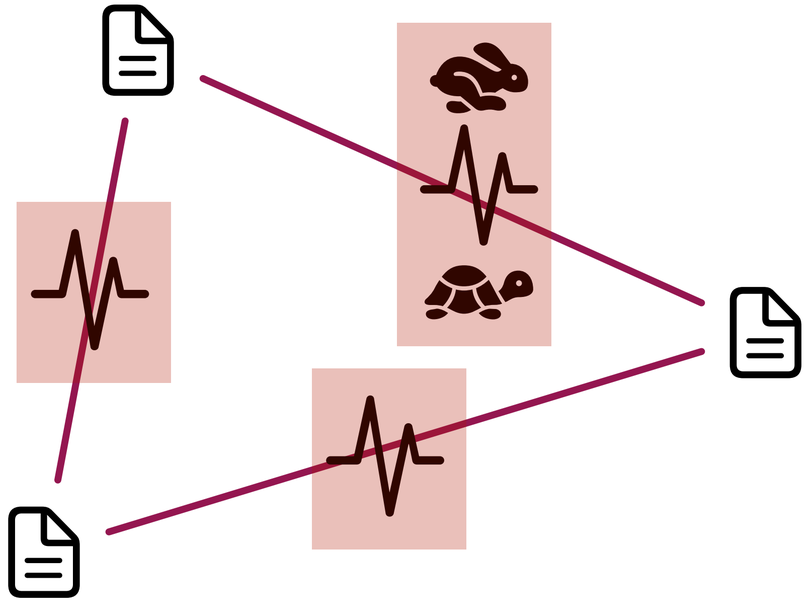
Challenge 2: Resilience despite asynchrony



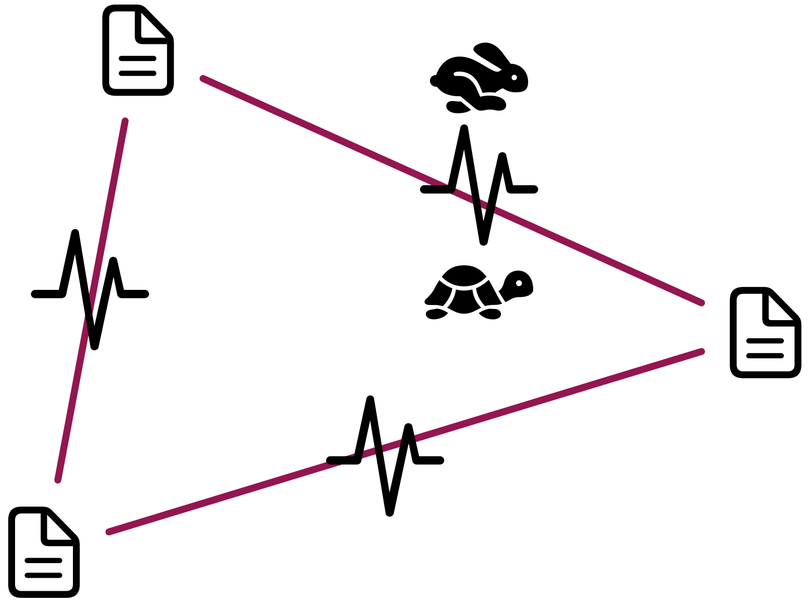
Challenge 2: Resilience despite asynchrony



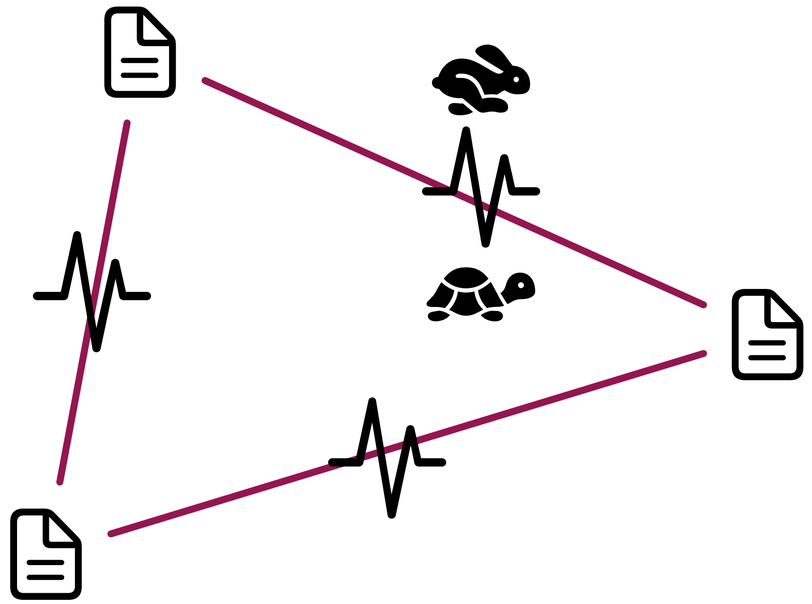
Challenge 2: Resilience despite asynchrony



Challenge 2: Resilience despite asynchrony



Challenge 2: Resilience despite asynchrony



Asynchronous consensus is robust to asynchrony, but rarely deployed

Towards more practical asynchronous consensus

Towards more practical asynchronous consensus

Asynchronous consensus typically uses a common coin

Towards more practical asynchronous consensus

Asynchronous consensus typically uses a common coin

Why? **Asynchronous consensus requires randomness for progress (FLP*)**

* Michael J Fischer, Nancy A Lynch, and Michael S Paterson. Impossibility of distributed consensus with one faulty process. *Journal of the ACM (JACM)*, 32(2):374–382, 1985.

Towards more practical asynchronous consensus

Asynchronous consensus typically uses a common coin

Why? **Asynchronous consensus requires randomness for progress (FLP*)**

Are common coins necessary for asynchronous consensus?

* Michael J Fischer, Nancy A Lynch, and Michael S Paterson. Impossibility of distributed consensus with one faulty process. *Journal of the ACM (JACM)*, 32(2):374–382, 1985.

Insight 2

No

Common coins not needed for asynchronous consensus (crash-fault setting)

Insight 2

No

Common coins not needed for asynchronous consensus (crash-fault setting)

Though no asymptotic improvement

Open question: Byzantine setting

Insight 2

No

Common coins not needed for asynchronous consensus (crash-fault setting)

Though no asymptotic improvement

Open question: Byzantine setting

First step towards more practical consensus

Resilient strongly-consistent coordination

Resilience to distant failures

Resilience to network asynchrony

Limit exposure

Resilient strongly-consistent coordination

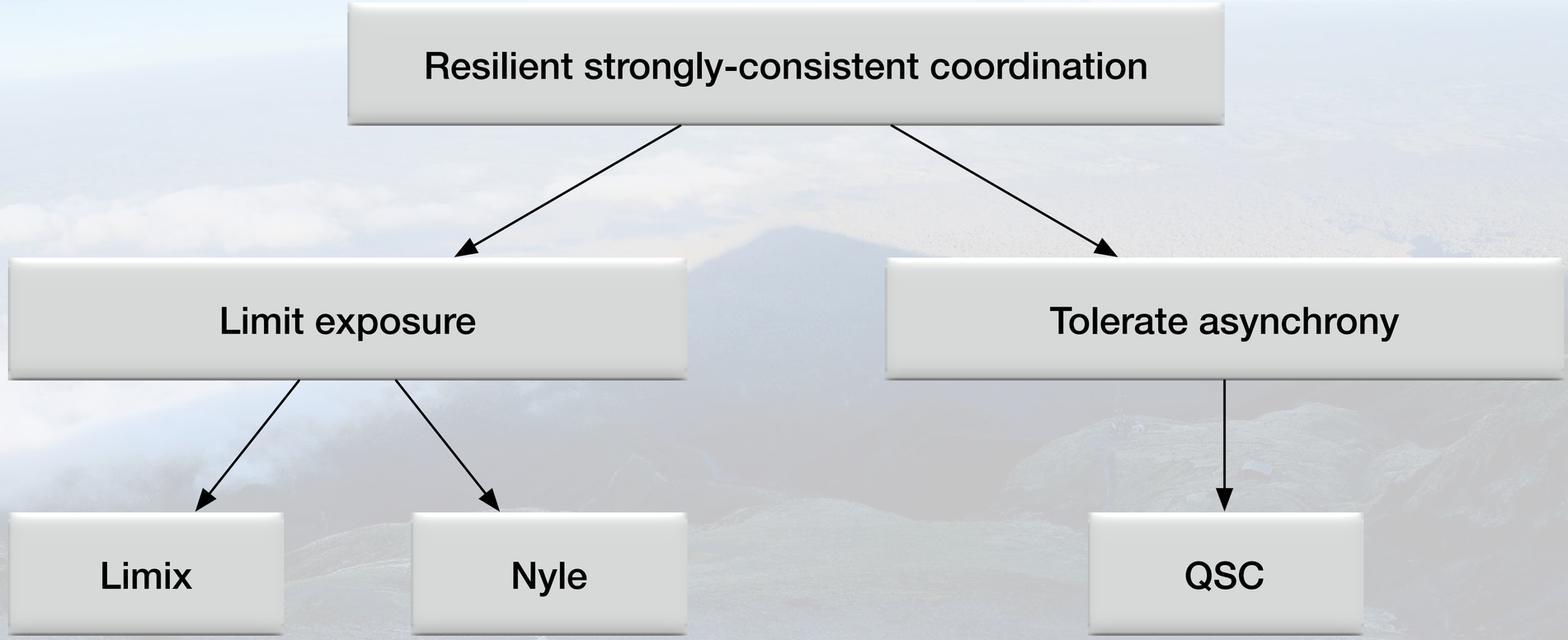
Resilience to distant failures

Resilience to network asynchrony

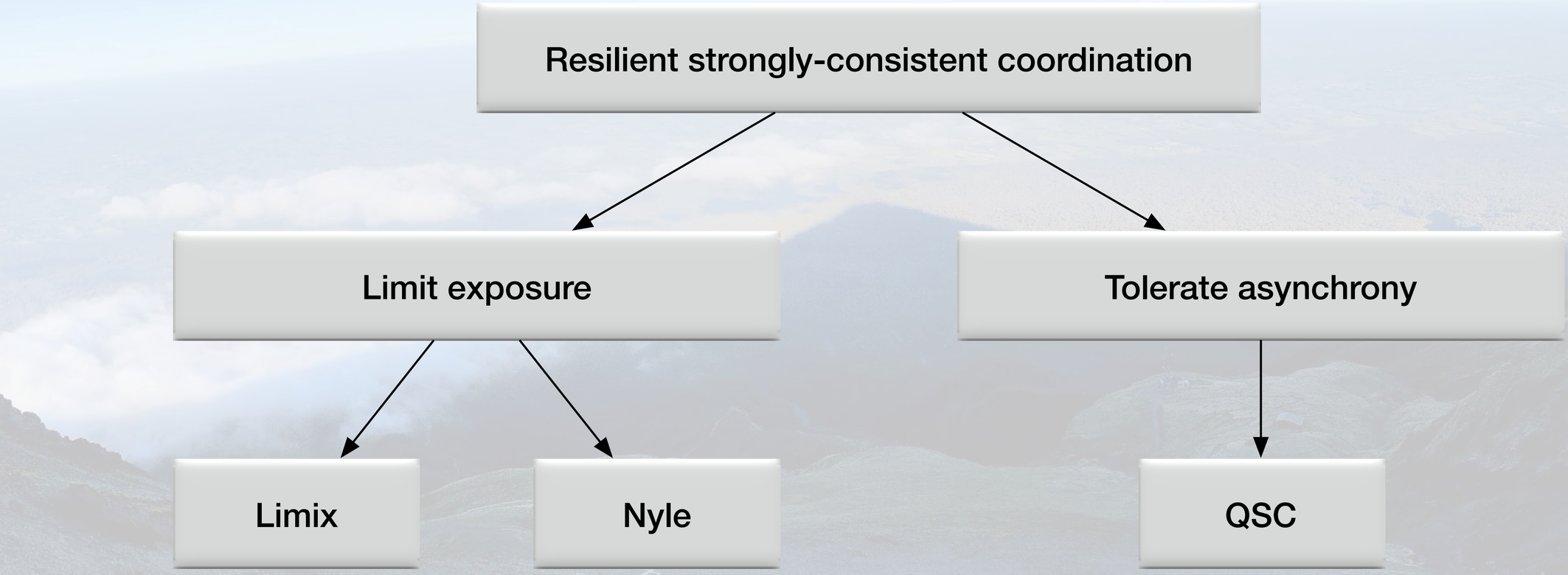
Limit exposure

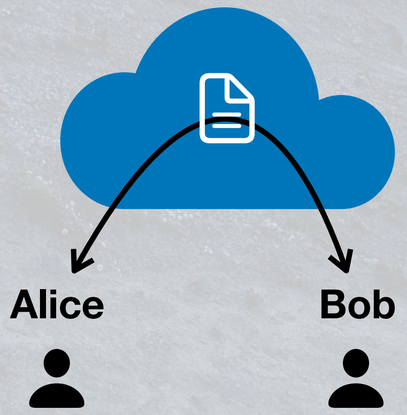
Common coin

Thesis roadmap



Thesis roadmap





Thesis roadmap

Resilient strongly-consistent coordination

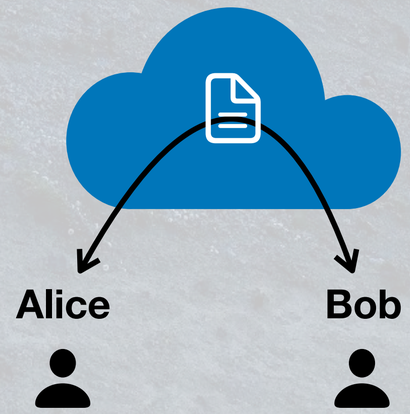
Limit exposure

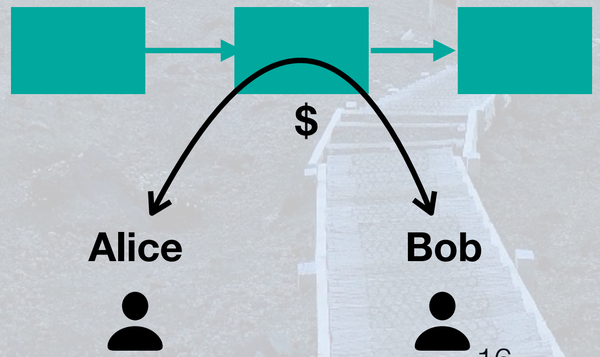
Tolerate asynchrony

Limix

Nyle

QSC





Thesis roadmap

Resilient strongly-consistent coordination

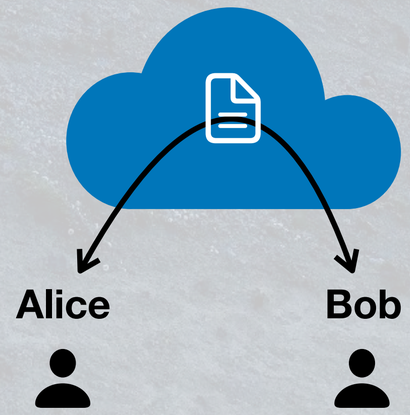
Limit exposure

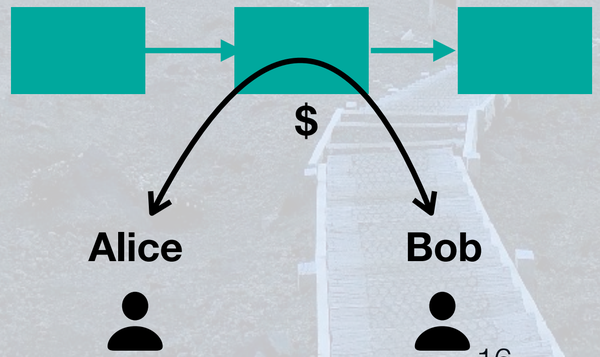
Tolerate asynchrony

Limix

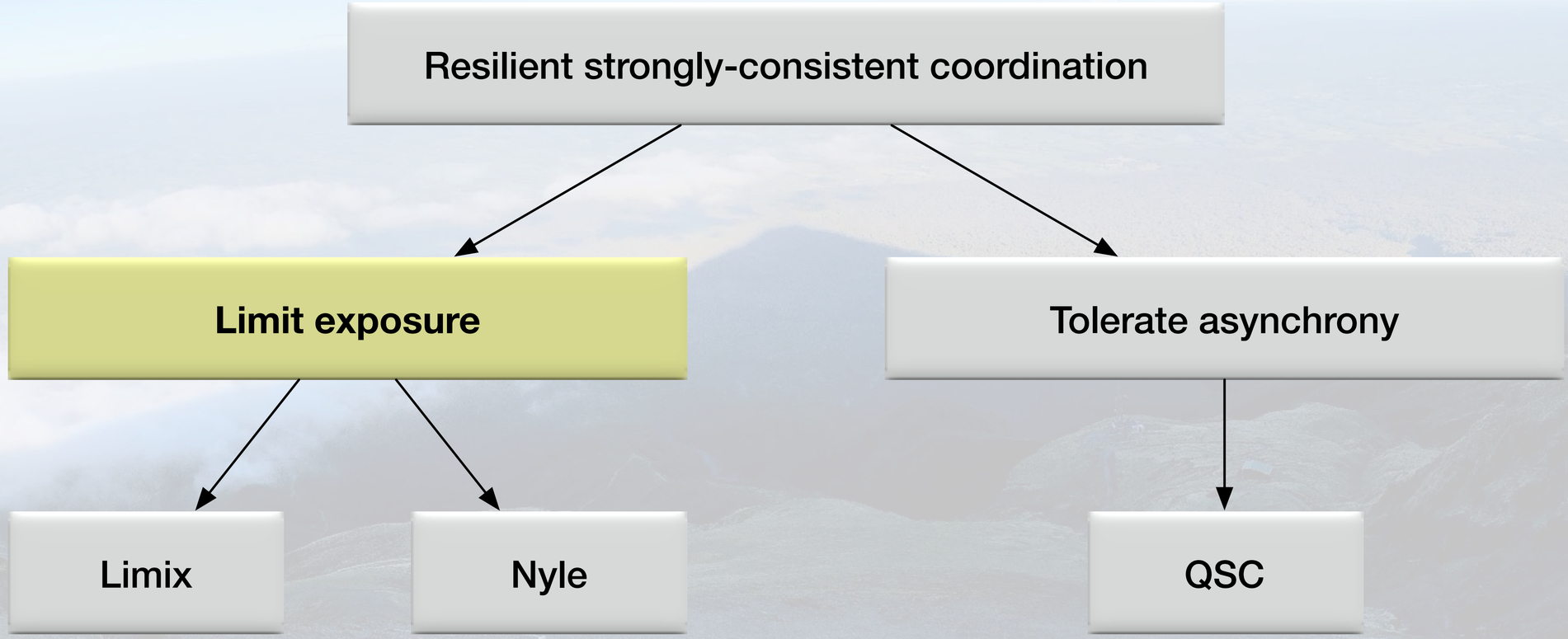
Nyle

QSC

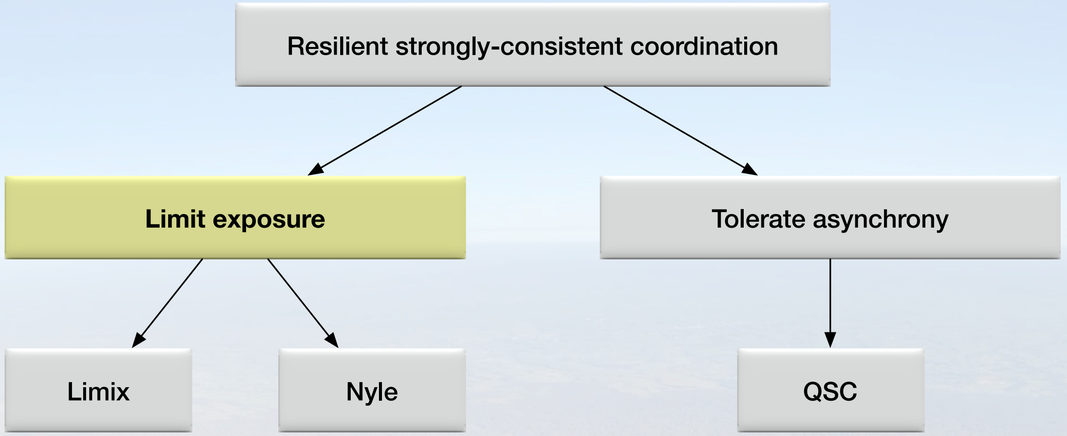




Thesis roadmap

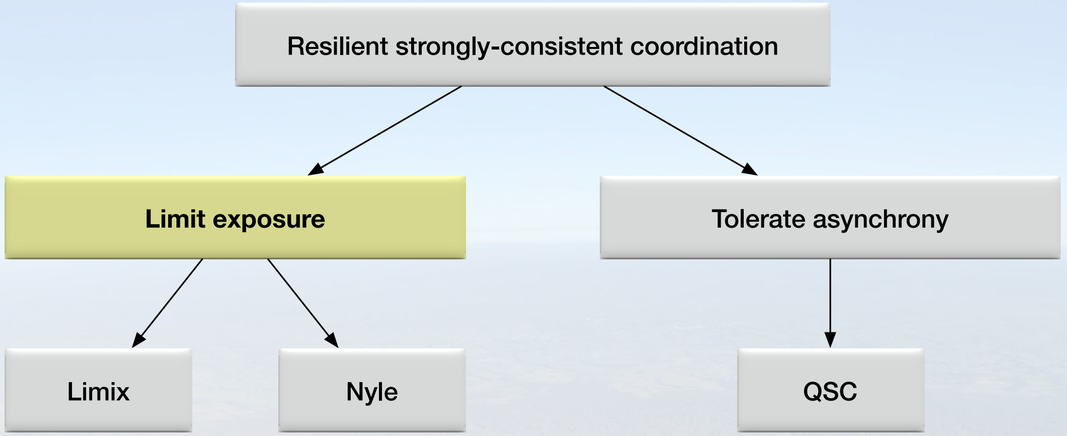


Limiting exposure roadmap



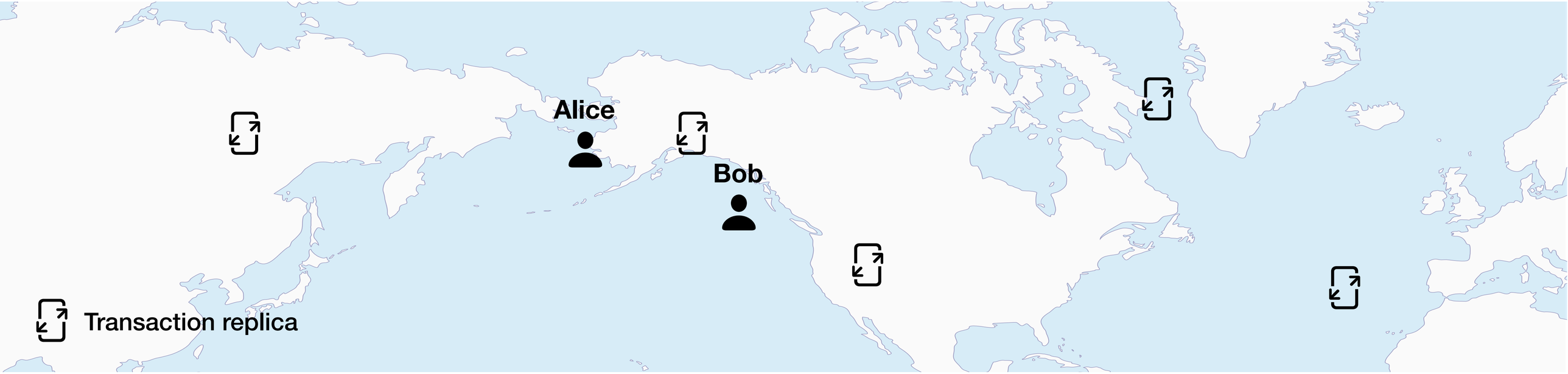
- Defining and measuring Lamport exposure
- Zoning: providing meaningful exposure guarantees
- Summary

Limiting exposure roadmap

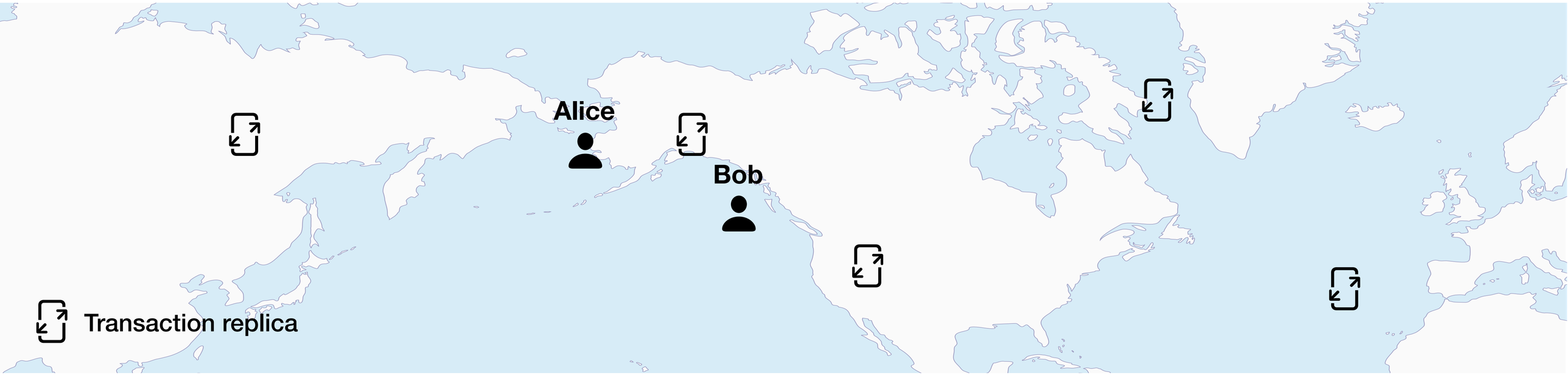


- **Defining and measuring Lamport exposure**
- Providing meaningful exposure guarantees
- Summary

Defining Lamport exposure

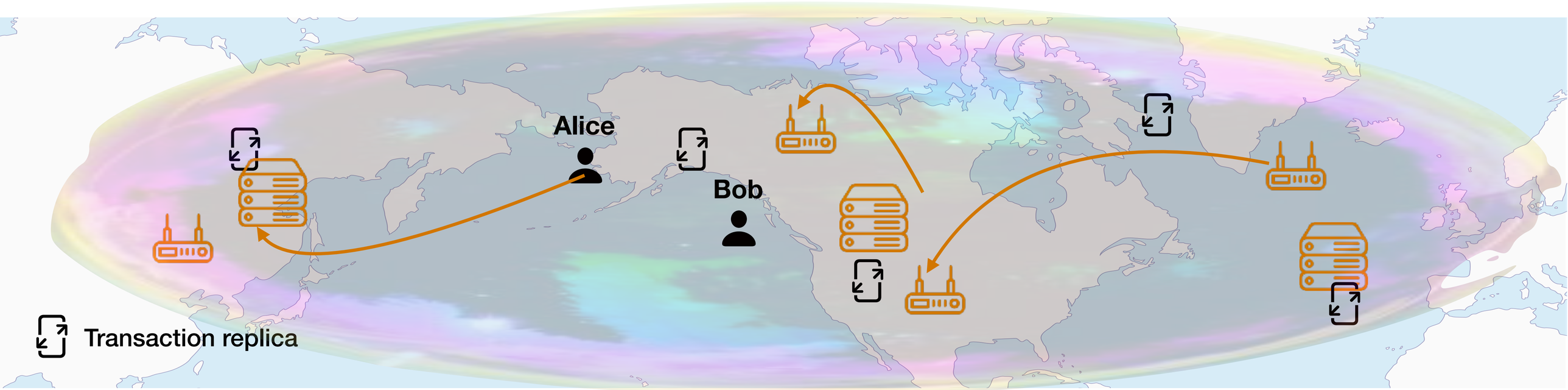


Defining Lamport exposure



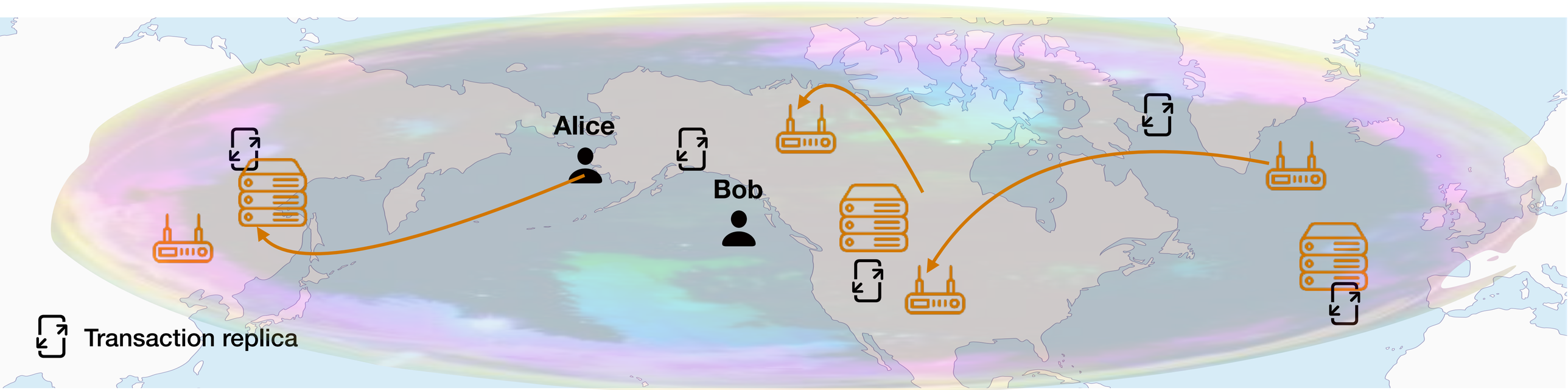
Lamport exposure for activity A of Alice
all distributed system components (servers, routers, network links, etc.)
whose failure **could** contribute to slowing or stopping activity A of a user.

Defining Lamport exposure



Lamport exposure for activity A of Alice
all distributed system components (servers, routers, network links, etc.)
whose failure **could** contribute to slowing or stopping activity A of a user.

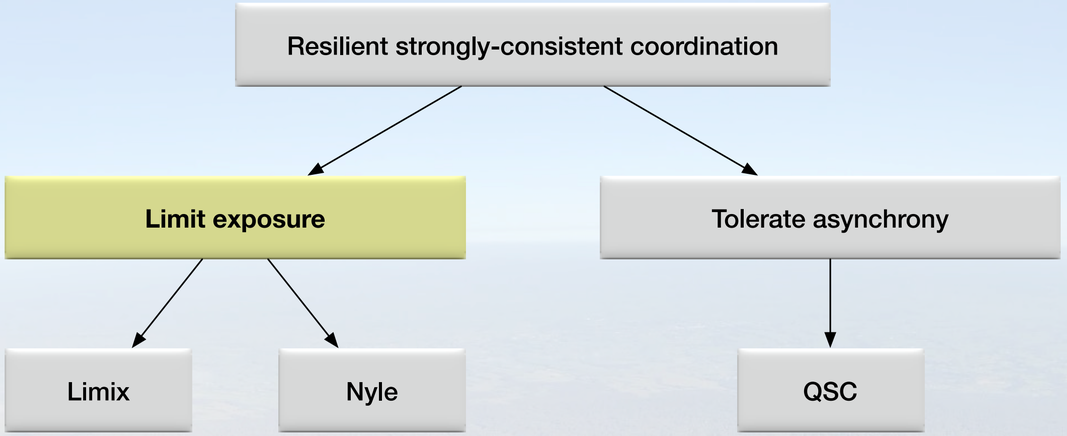
Defining Lamport exposure



Lamport exposure for activity A of Alice
all distributed system components (servers, routers, network links, etc.)
whose failure **could** contribute to slowing or stopping activity A of a user.

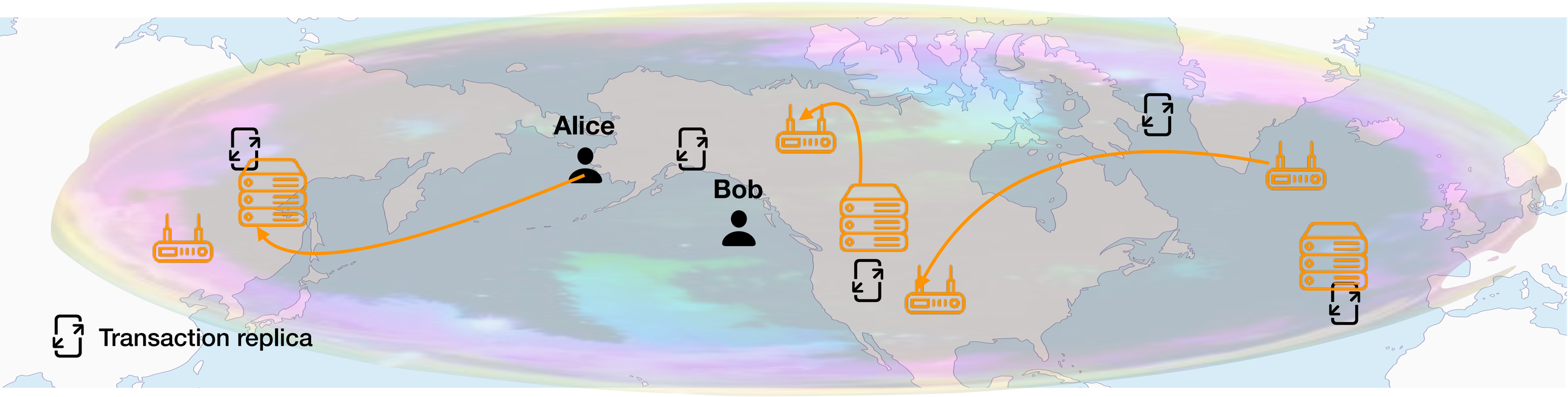
Global system: Lamport exposure is some global set the user is unaware of

Limiting exposure roadmap



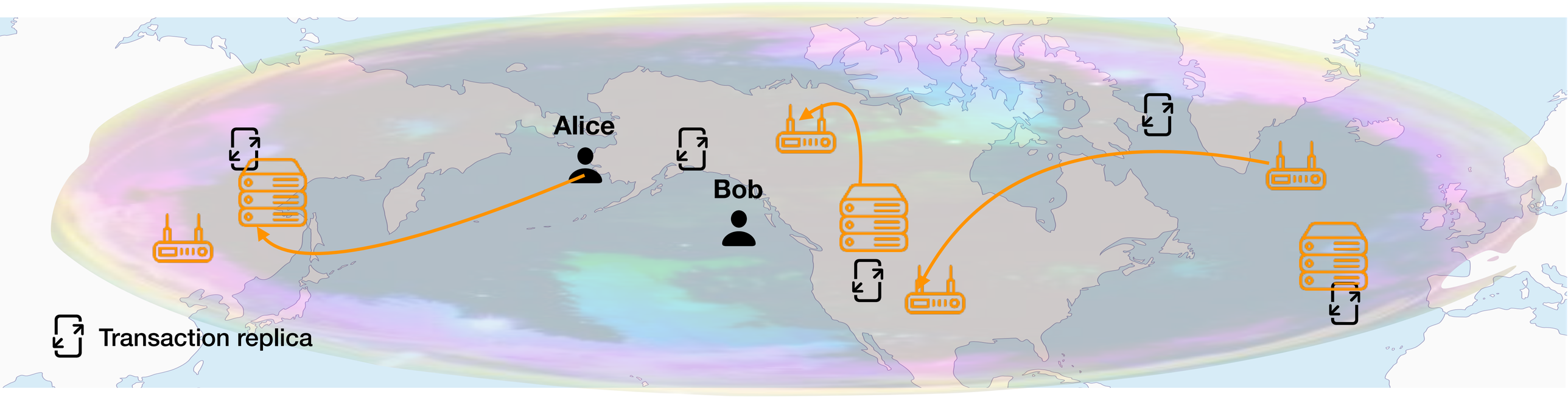
- Defining and measuring Lamport exposure
- **Providing meaningful exposure guarantees**
  - **Goals**
  - Strawmen
  - Jurisdiction-based zoning
  - Metric-based zoning
- Summary

Limit exposure & preserve global manageability



Goals:

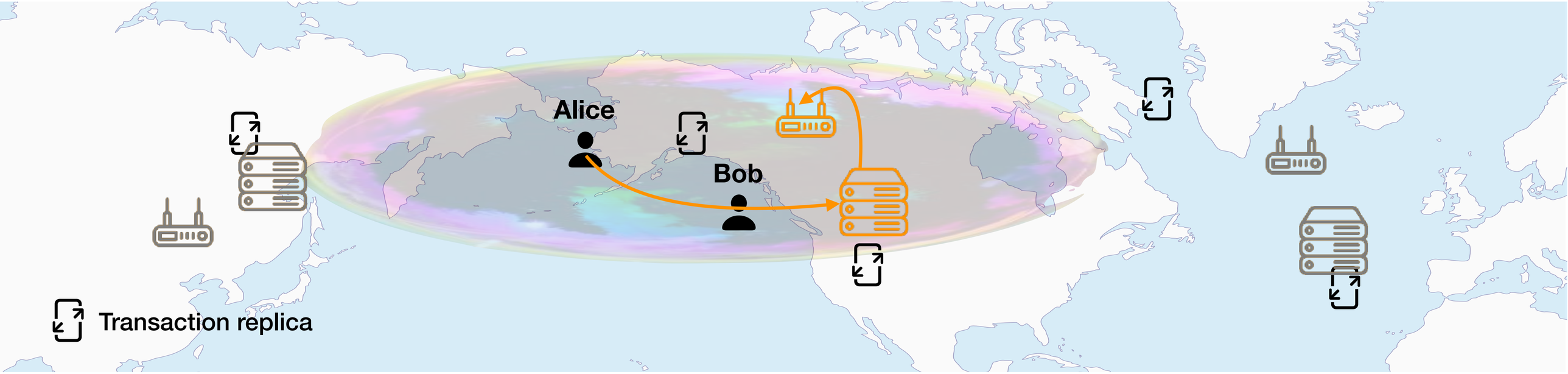
Limit exposure & preserve global manageability



Goals:

Alice has a lower exposure as Bob gets closer

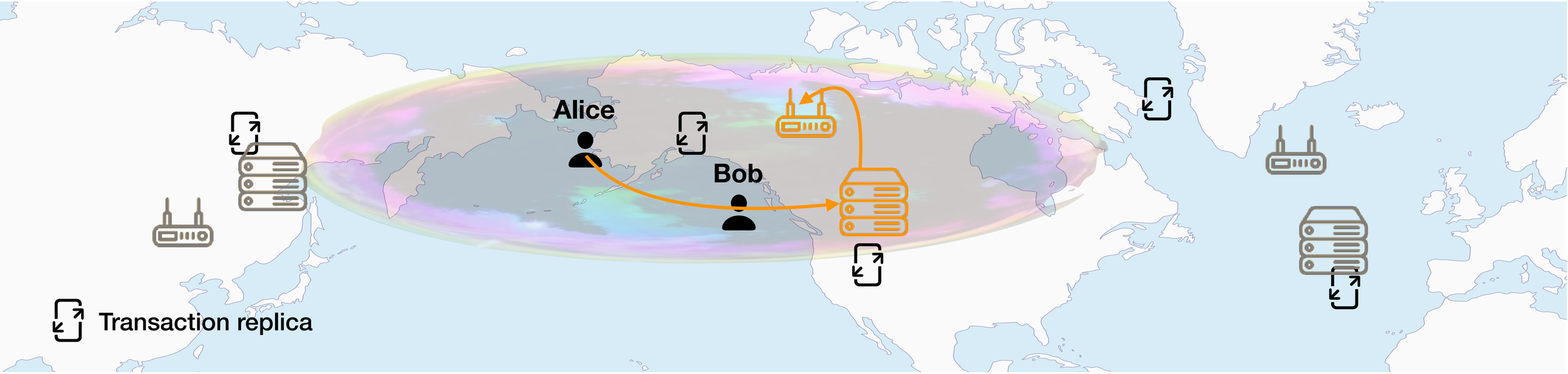
Limit exposure & preserve global manageability



Goals:

Alice has a lower exposure as Bob gets closer

Limit exposure & preserve global manageability

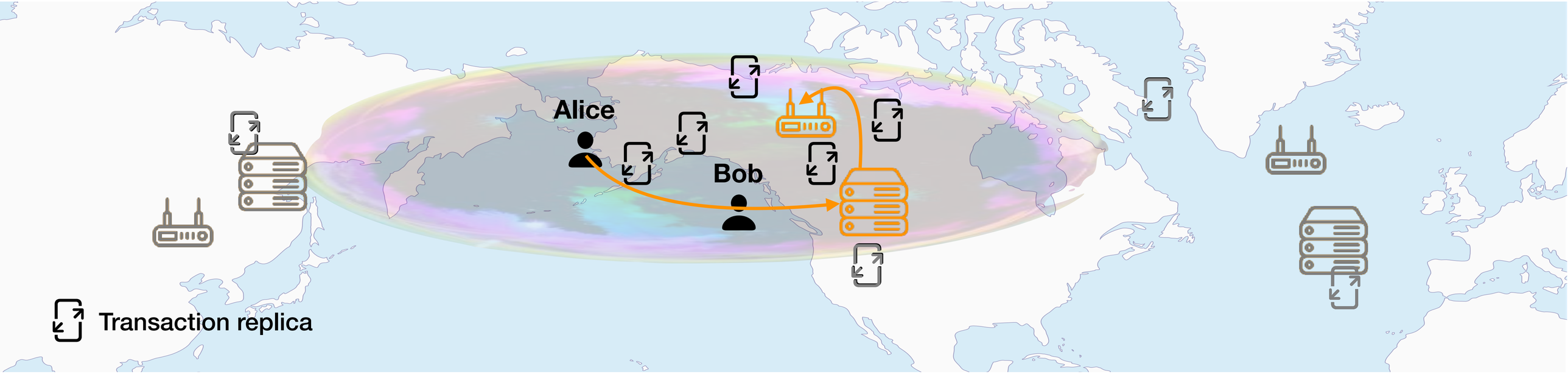


Goals:

Alice has a lower exposure as Bob gets closer

Limiting exposure does not reduce durability

Limit exposure & preserve global manageability

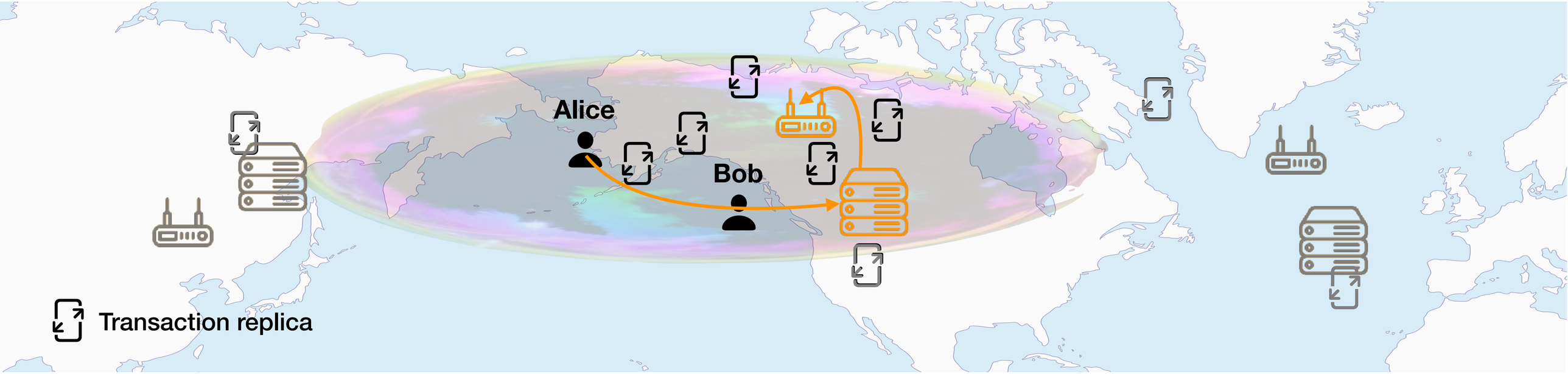


Goals:

Alice has a lower exposure as Bob gets closer

Limiting exposure does not reduce durability

Limit exposure & preserve global manageability



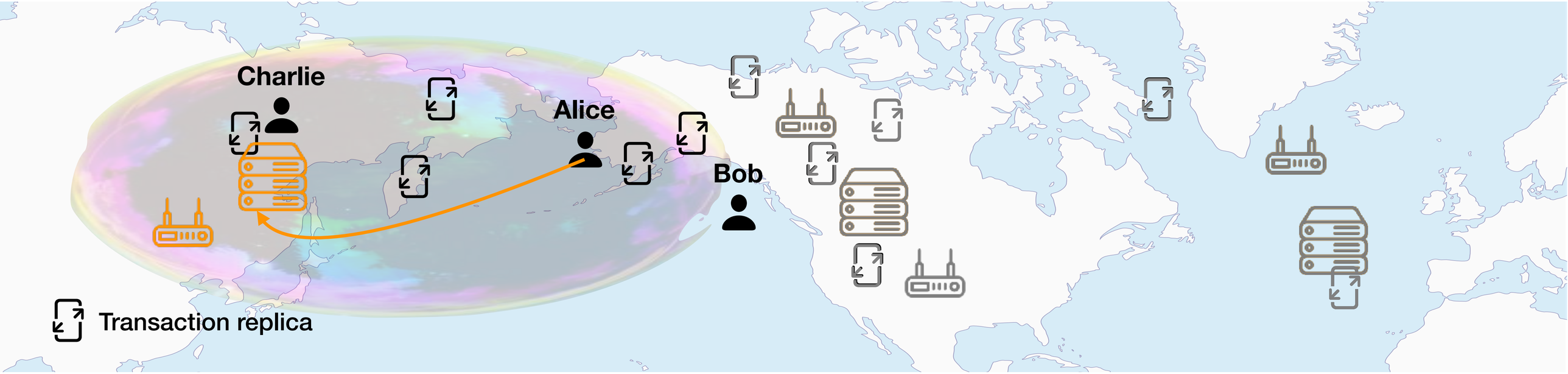
Goals:

Alice has a lower exposure as Bob gets closer

Limiting exposure does not reduce durability

Exposure adjusts to new interactions and users changing location

Limit exposure & preserve global manageability



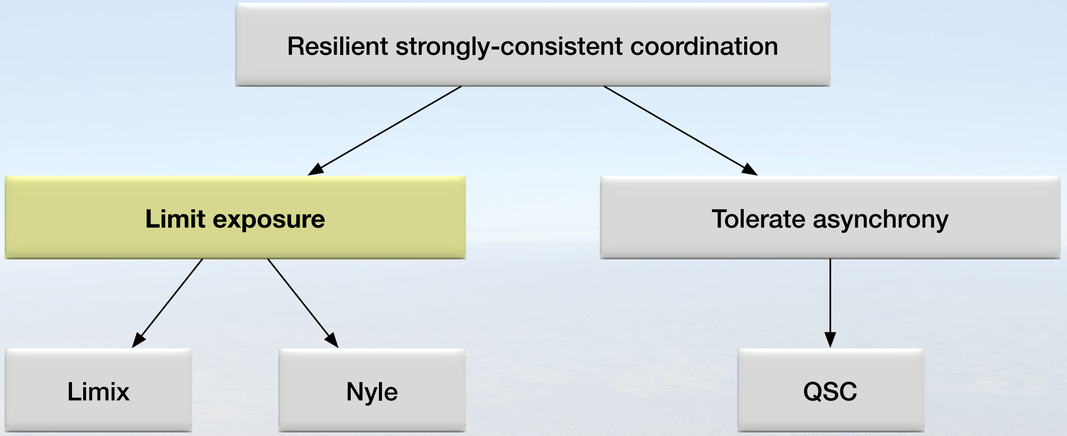
Goals:

Alice has a lower exposure as Bob gets closer

Limiting exposure does not reduce durability

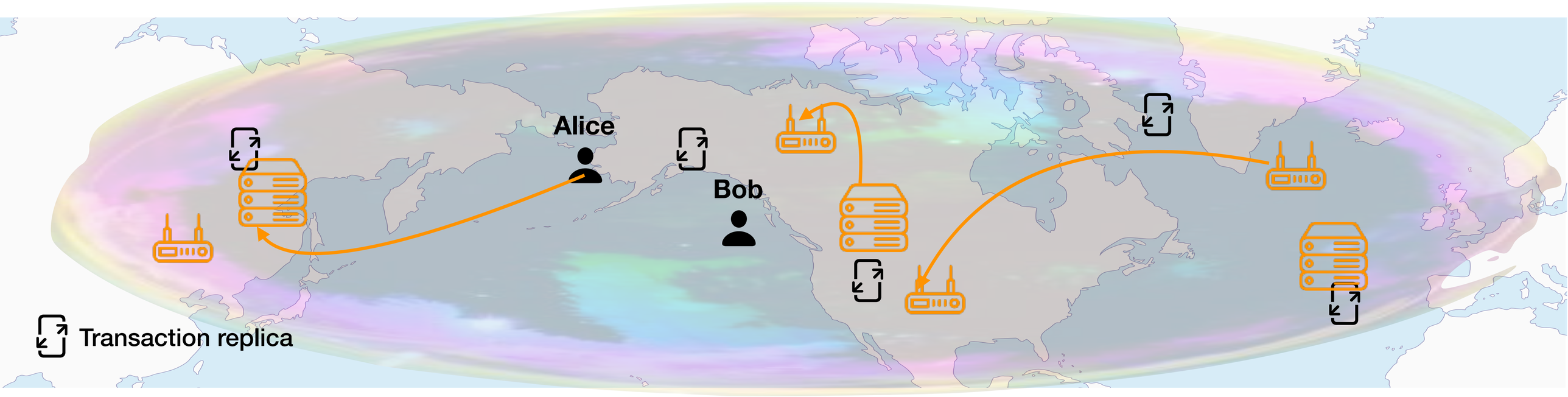
Exposure adjusts to new interactions and users changing location

Limiting exposure roadmap

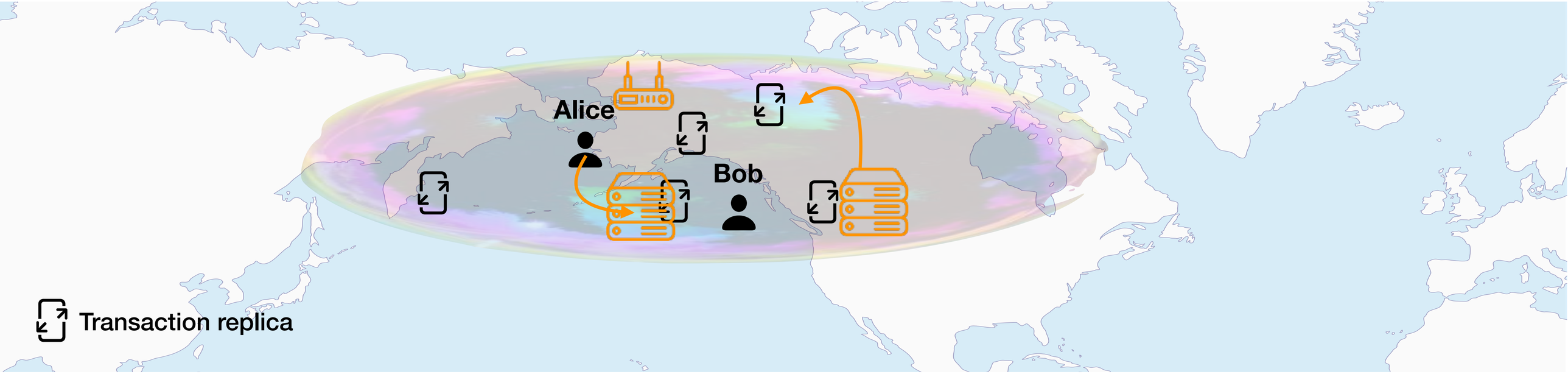


- Defining Lamport exposure
- **Providing meaningful exposure guarantees**
  - Goals
  - **Strawmen**
  - Jurisdiction-based zoning
  - Metric-based zoning
- Summary

Strawman 1: localised infrastructure

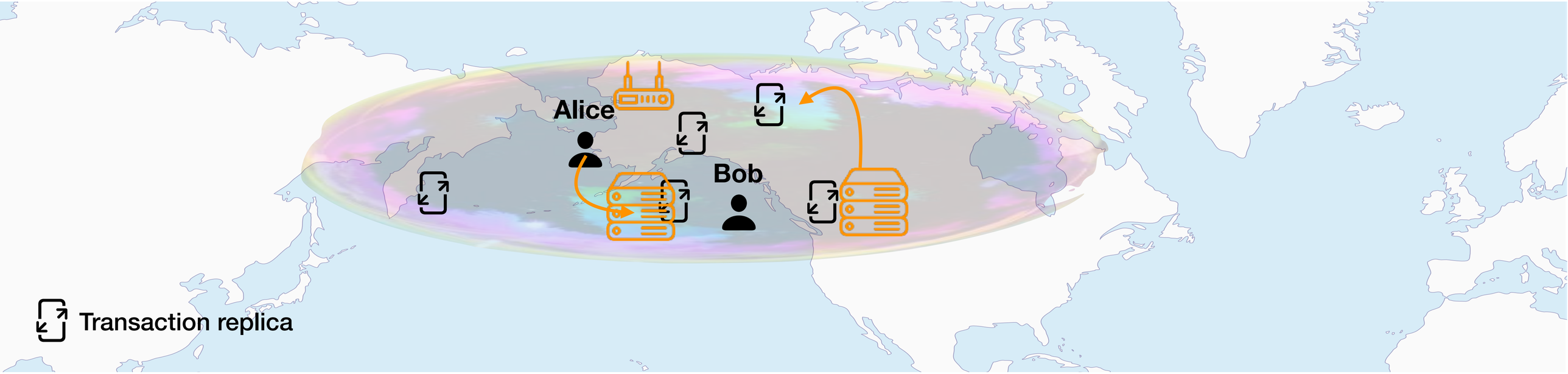


Strawman 1: localised infrastructure



A localized deployment, e.g. local ledger, instead of globalised one

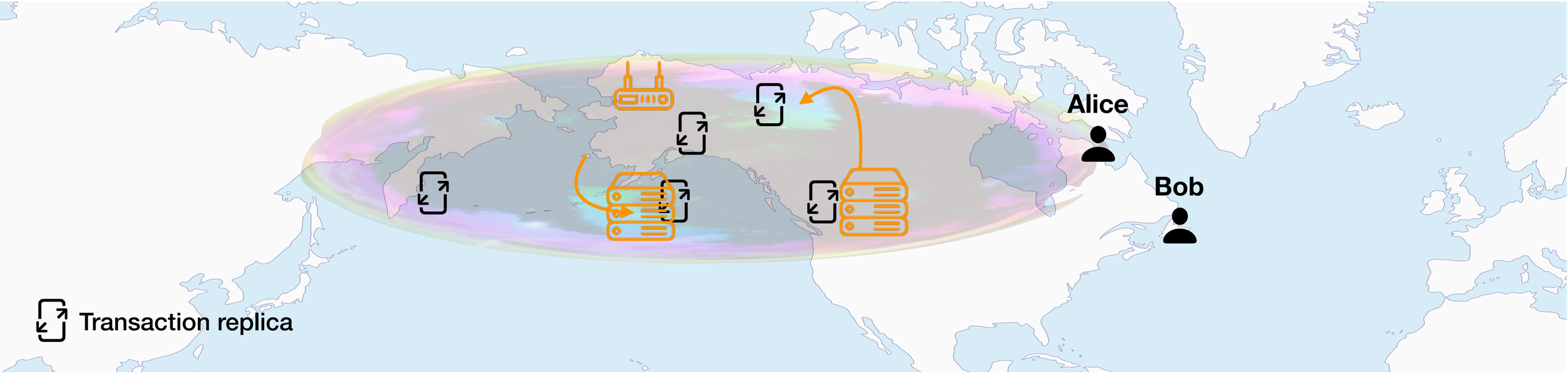
Strawman 1: localised infrastructure



A localized deployment, e.g. local ledger, instead of globalised one

✓ **Resilience: Lower exposure for local users**

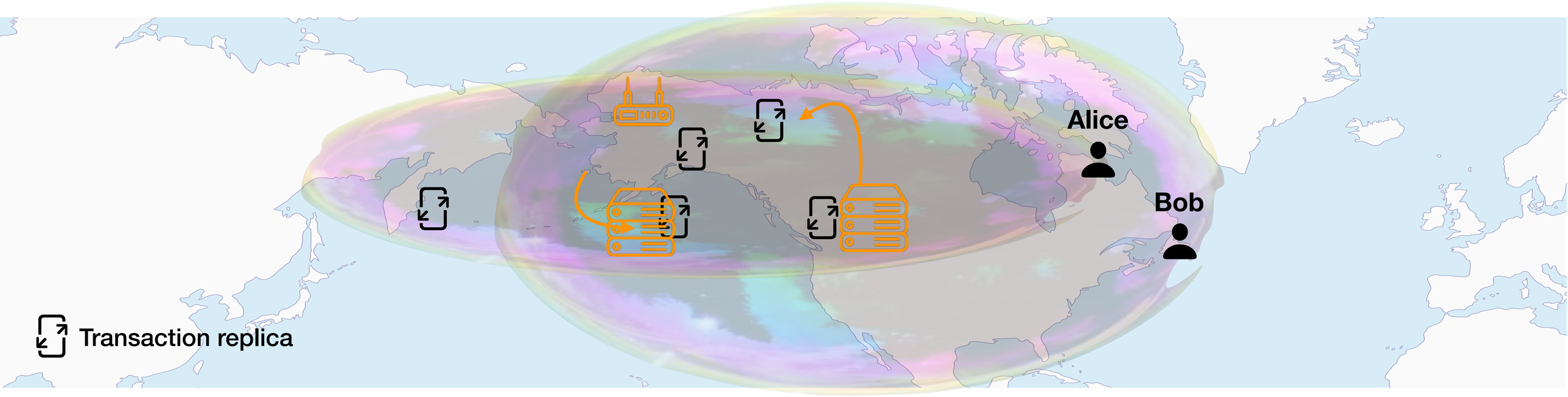
Strawman 1: localised infrastructure



A localized deployment, e.g. local ledger, instead of globalised one

✓ **Resilience: Lower exposure for local users**

Strawman 1: localised infrastructure



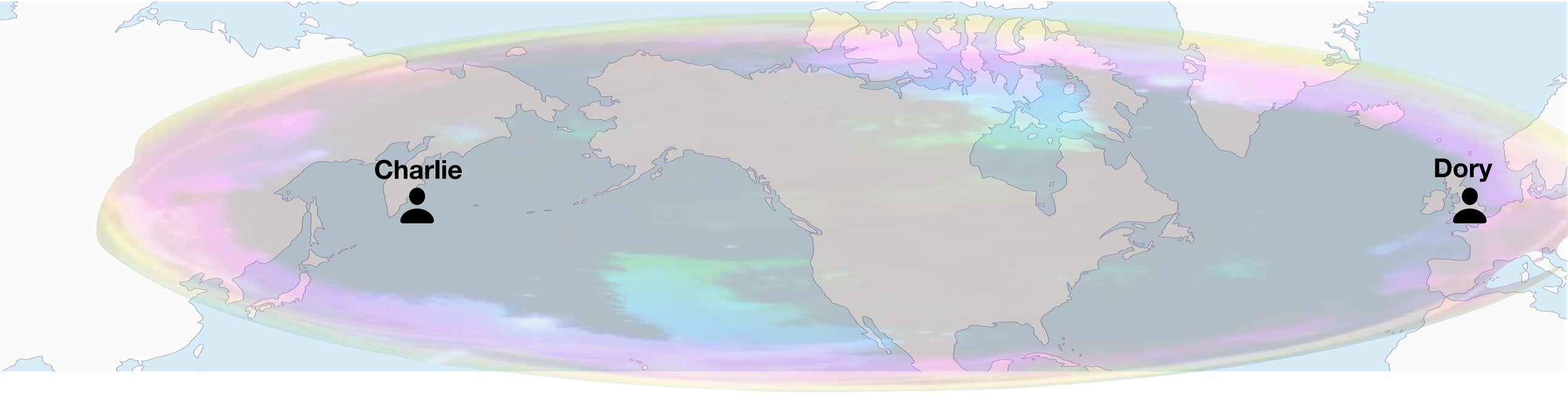
A localized deployment, e.g. local ledger, instead of globalised one

✓ **Resilience: Lower exposure for local users**

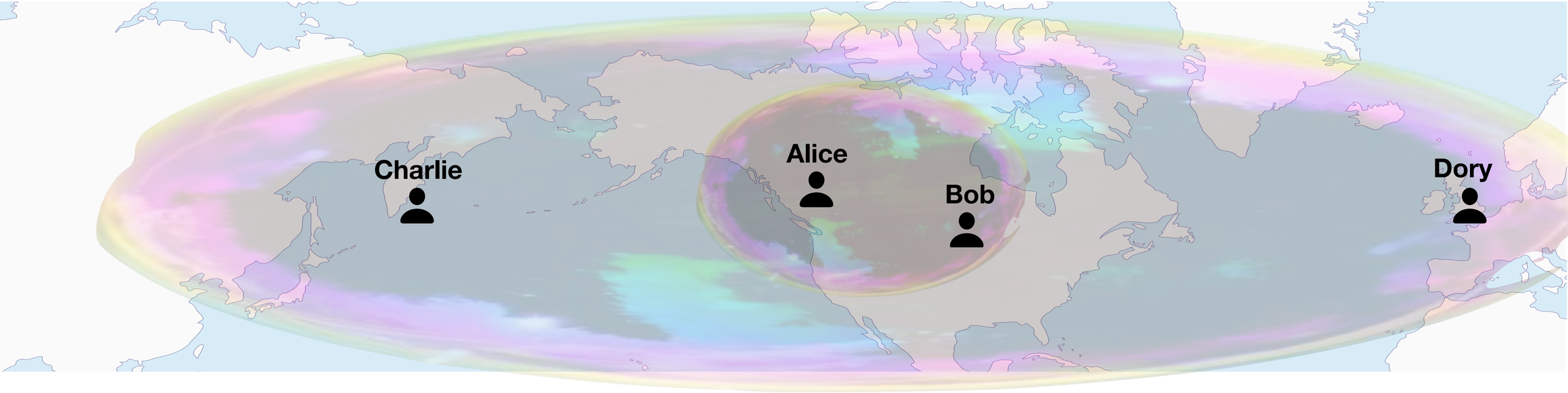
✗ **No global manageability: migrating users have remote dependencies**

Strawman 2: many zones

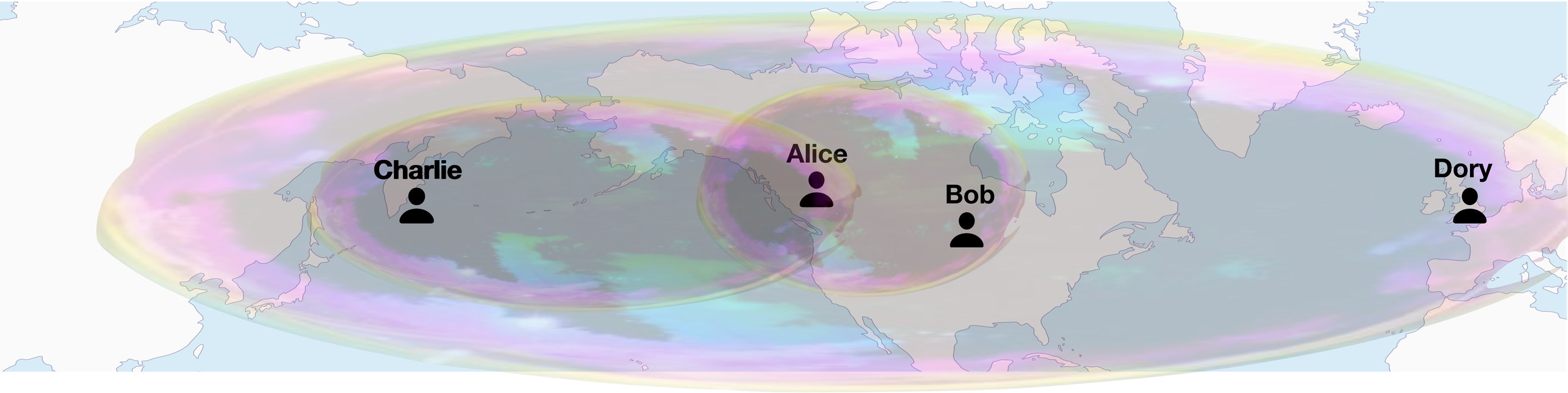
Strawman 2: many zones



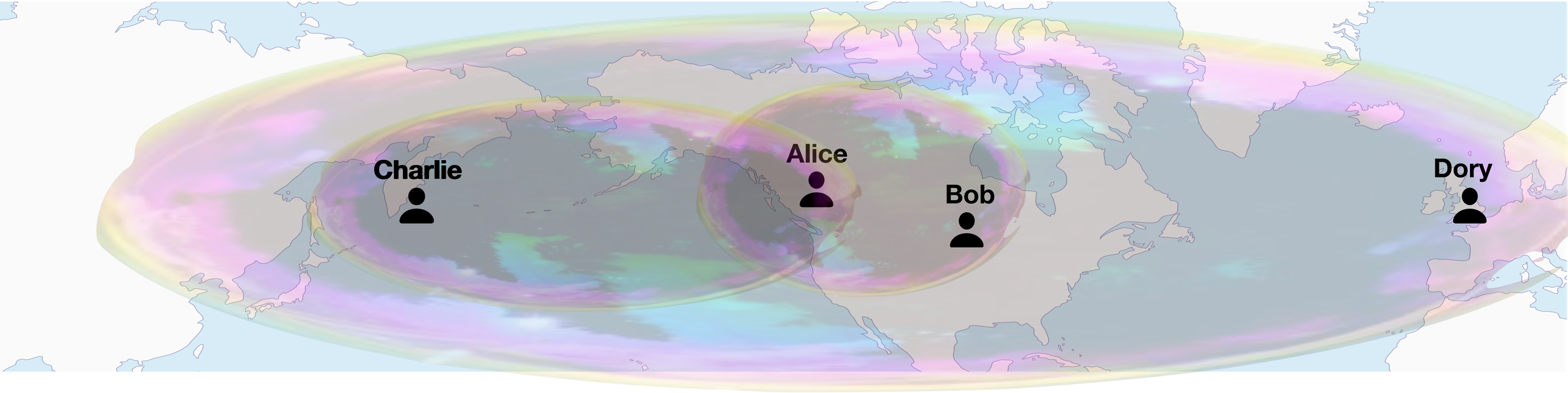
Strawman 2: many zones



Strawman 2: many zones

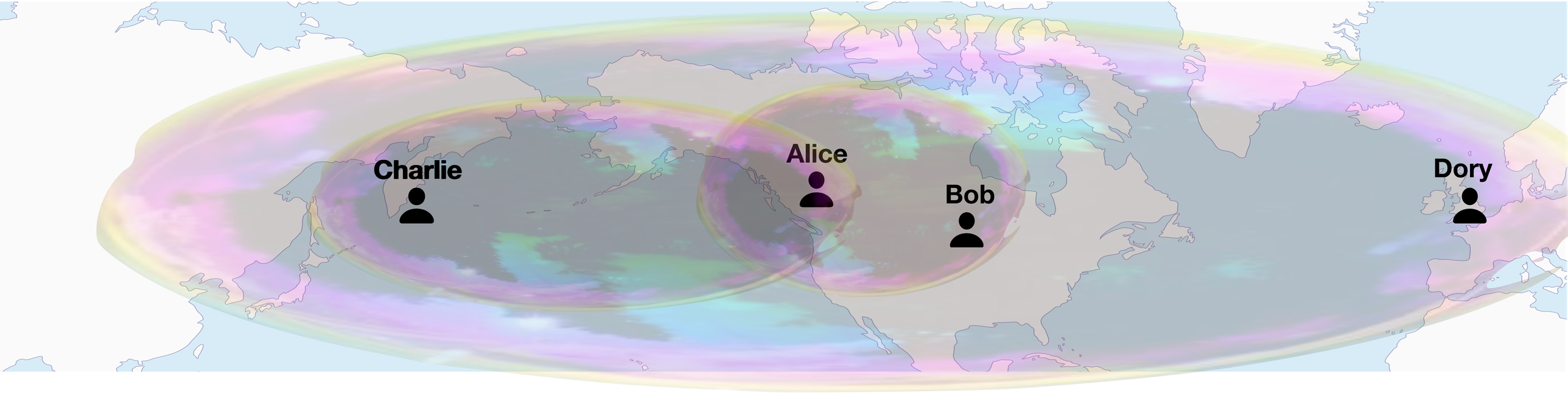


Strawman 2: many zones



✓ **Resilience: shield users from failures outside the zone**

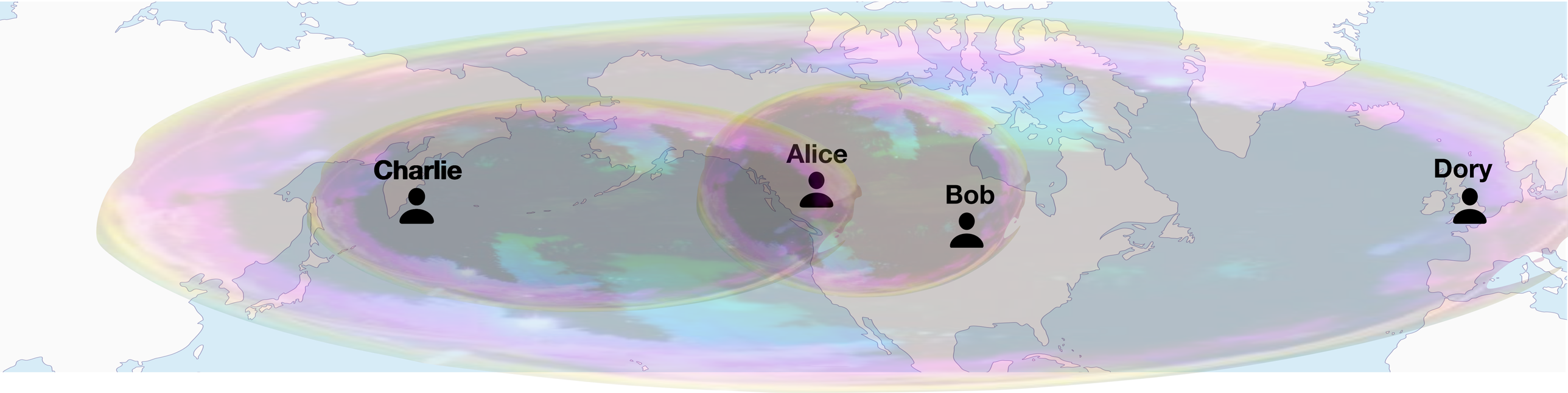
Strawman 2: many zones



✓ **Resilience**: shield users from failures outside the zone

✓ **Global manageability**: dependencies follow usage pattern

Strawman 2: many zones



✓ **Resilience**: shield users from failures outside the zone

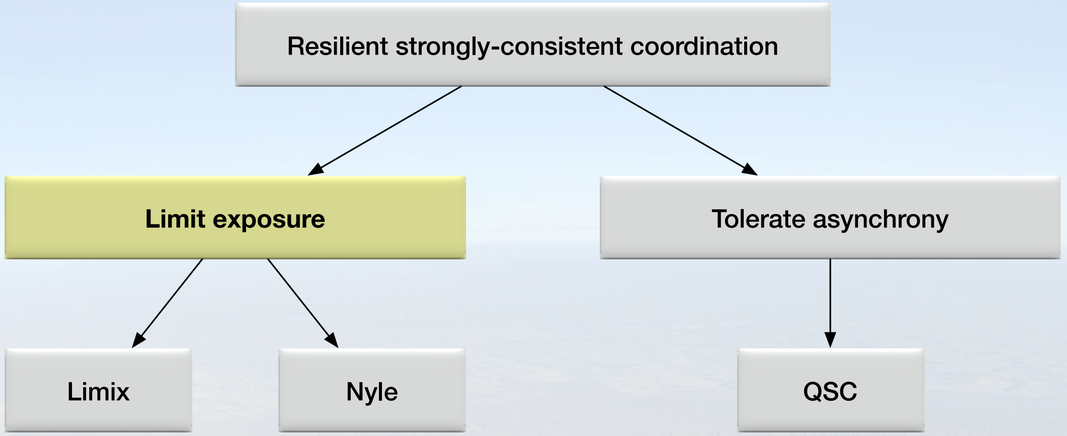
✓ **Global manageability**: dependencies follow usage pattern

✗ **Overhead**: replication inside each of the many zones

Insight

User interactions that the ledger aims to support determine the zoning policy

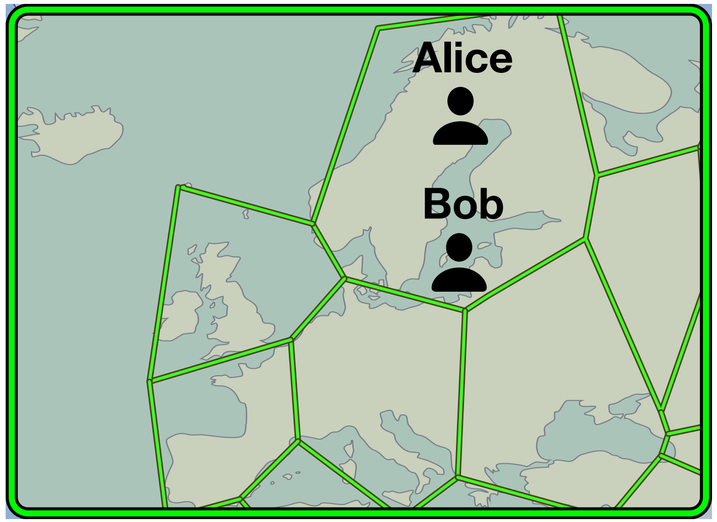
Limiting exposure roadmap



- Defining Lamport exposure
- **Providing meaningful exposure guarantees**
  - Goals
  - Strawmen
  - **Jurisdiction-based zoning**
  - Metric-based zoning
- Summary

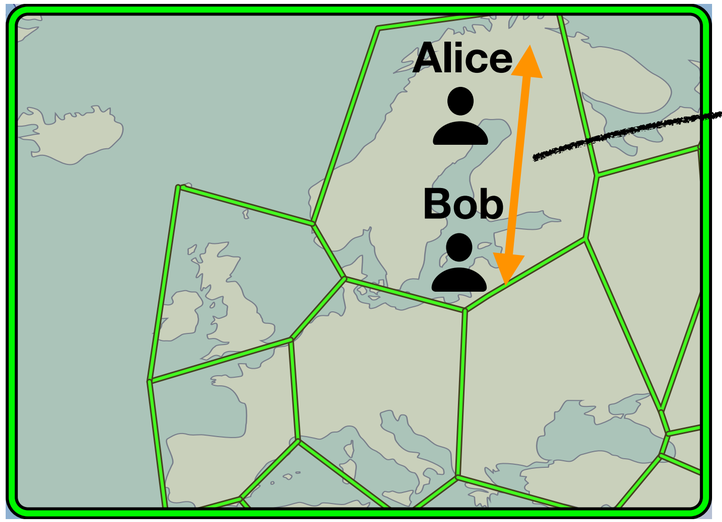
Jurisdiction-based zoning

- Administrative or legal boundaries (e.g., inside/outside the EU)
  Use case: data sovereignty, enforcing legal frameworks for transactions



Jurisdiction-based zoning

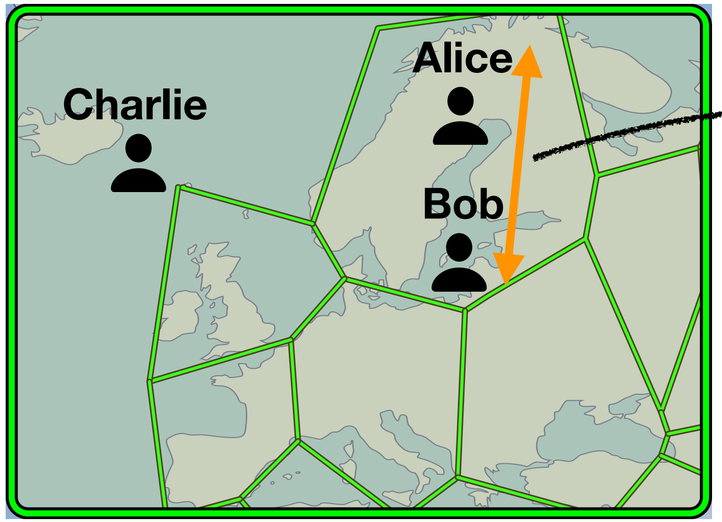
- Administrative or legal boundaries (e.g., inside/outside the EU)
  Use case: data sovereignty, enforcing legal frameworks for transactions



Measures **Alice's** Lamport exposure
to failures or slowdowns
for transactions with **Bob**

Jurisdiction-based zoning

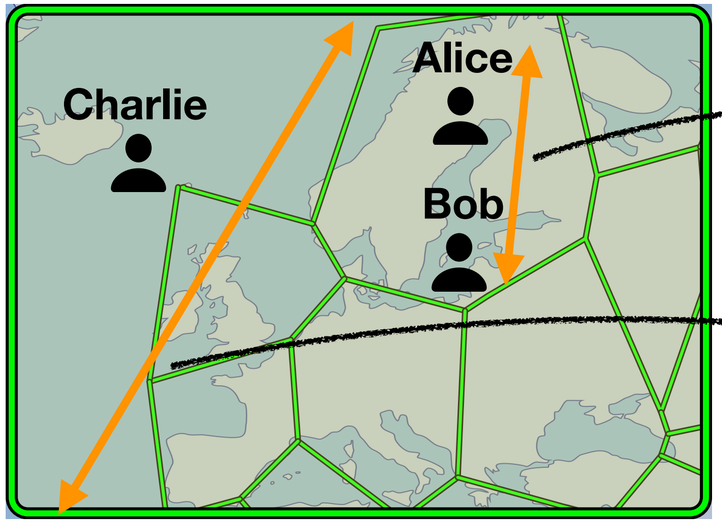
- Administrative or legal boundaries (e.g., inside/outside the EU)
  Use case: data sovereignty, enforcing legal frameworks for transactions



Measures **Alice's** Lamport exposure
to failures or slowdowns
for transactions with **Bob**

Jurisdiction-based zoning

- Administrative or legal boundaries (e.g., inside/outside the EU)
  Use case: data sovereignty, enforcing legal frameworks for transactions

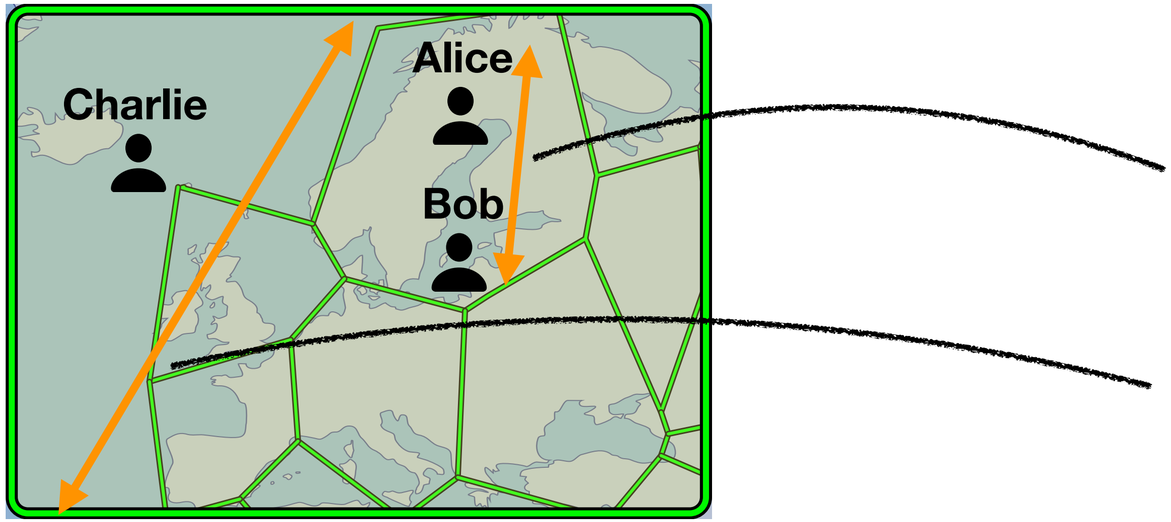


Measures **Alice's** Lamport exposure
to failures or slowdowns
for transactions with **Bob**

Measures **Alice's** Lamport exposure
to failures or slowdowns
for transactions with **Charlie**

Jurisdiction-based zoning

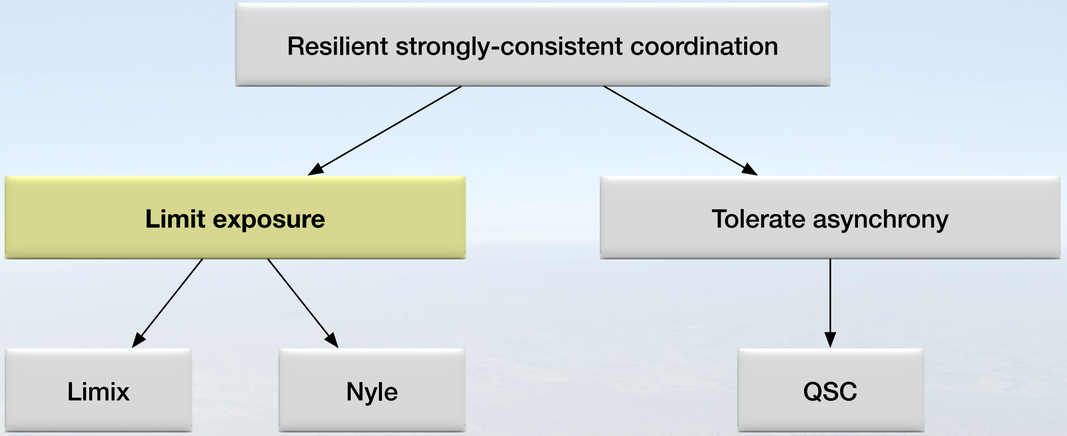
- Administrative or legal boundaries (e.g., inside/outside the EU)
  Use case: data sovereignty, enforcing legal frameworks for transactions
- Can be used in a pay-as-you-go fashion



Measures **Alice's** Lamport exposure
to failures or slowdowns
for transactions with **Bob**

Measures **Alice's** Lamport exposure
to failures or slowdowns
for transactions with **Charlie**

Limiting exposure roadmap

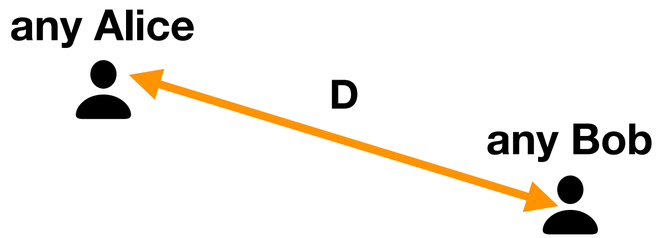


- Defining Lamport exposure
- **Providing meaningful exposure guarantees**
  - Goals
  - Strawmen
  - Jurisdiction-based zoning
  - **Metric-based zoning**
- Summary

Metric-based zoning

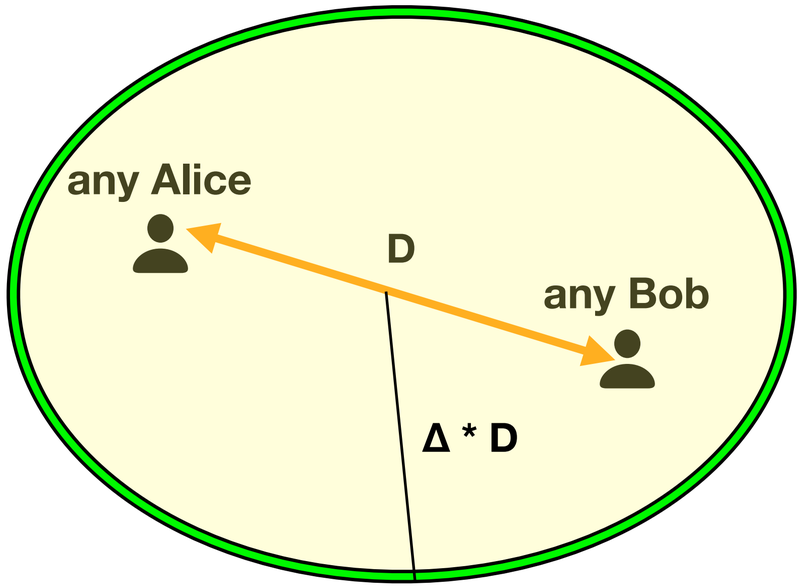
Metric-based zoning

- **Dynamic local and global interactions**



Metric-based zoning

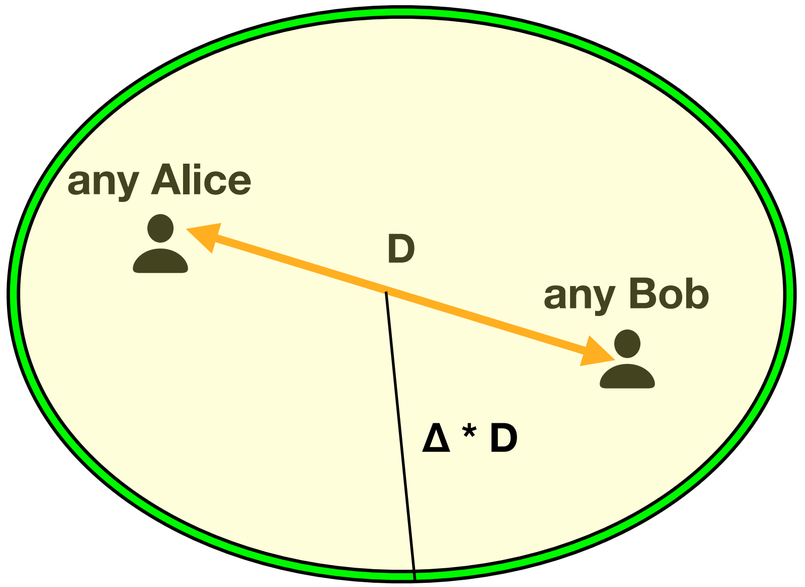
- **Dynamic local and global interactions**
  - Limit exposure to a small Δ around **any two users**



Metric-based zoning

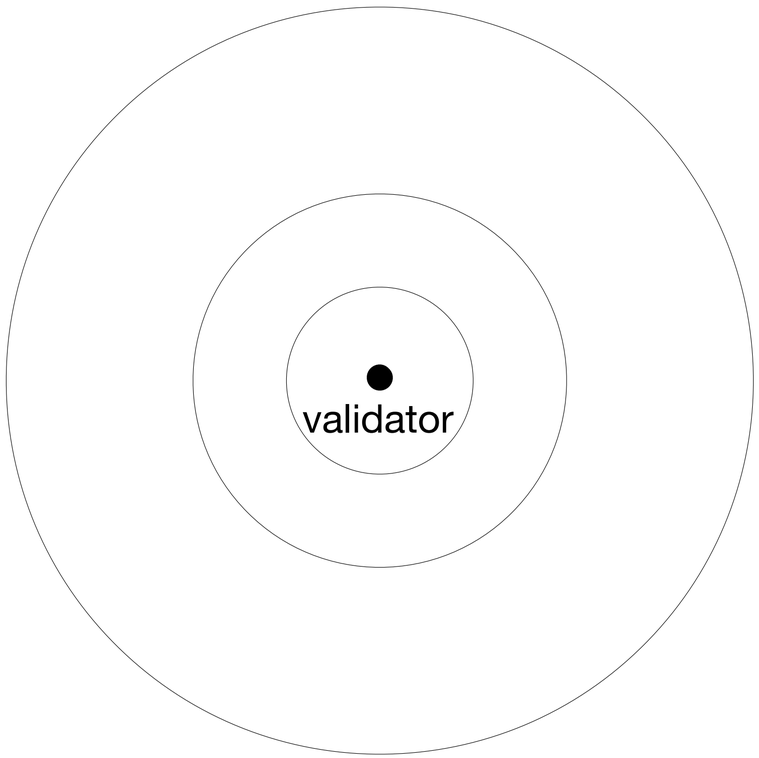
- **Dynamic local and global interactions**
  - Limit exposure to a small Δ around **any two users**

- **Metrics**
  - Round-trip time (RTT)
    Use case: low latency transactions
  - Geographic-straight line distance
    Use case: community-focused deployment
  - Network hop count
    Use case: paths with fewer hops are more resilient



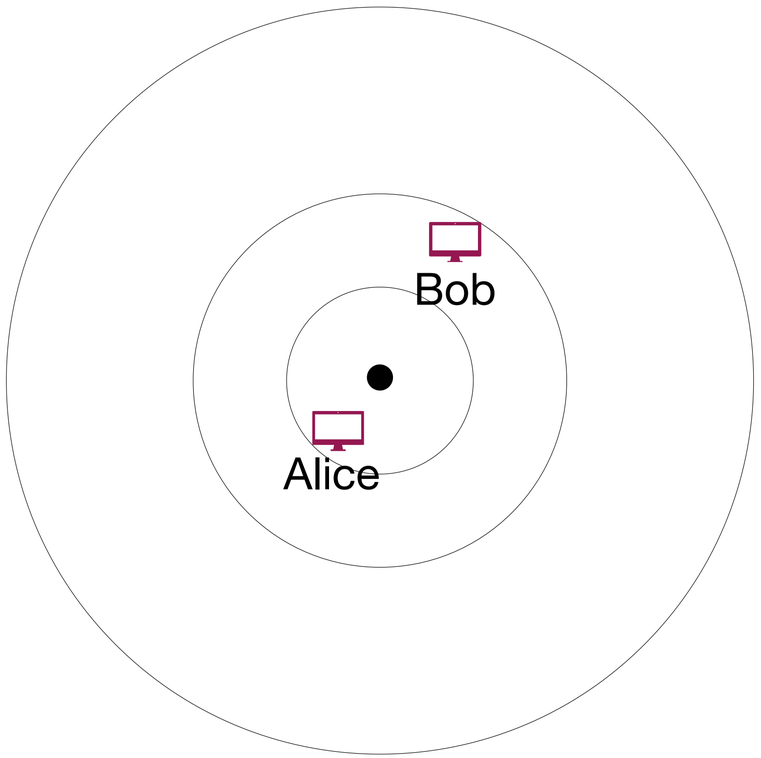
Strawman

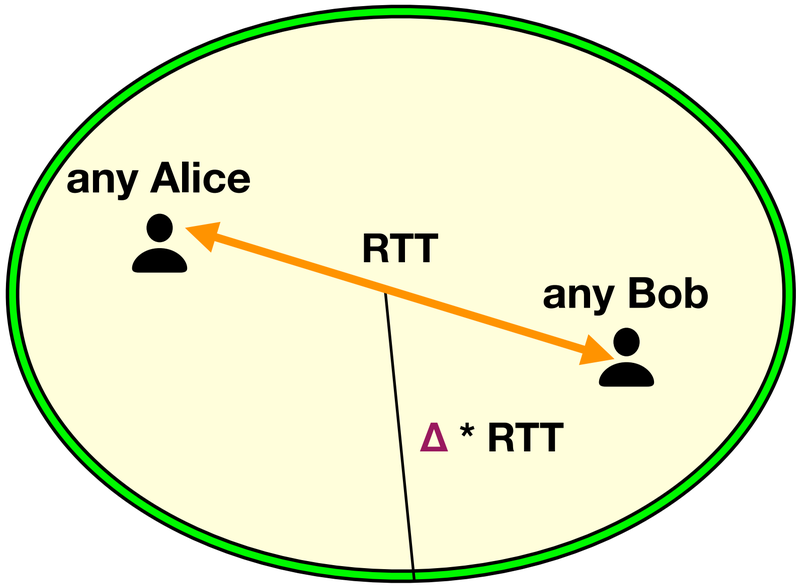
- Builds exponentially-increasing zones around each validator



Strawman

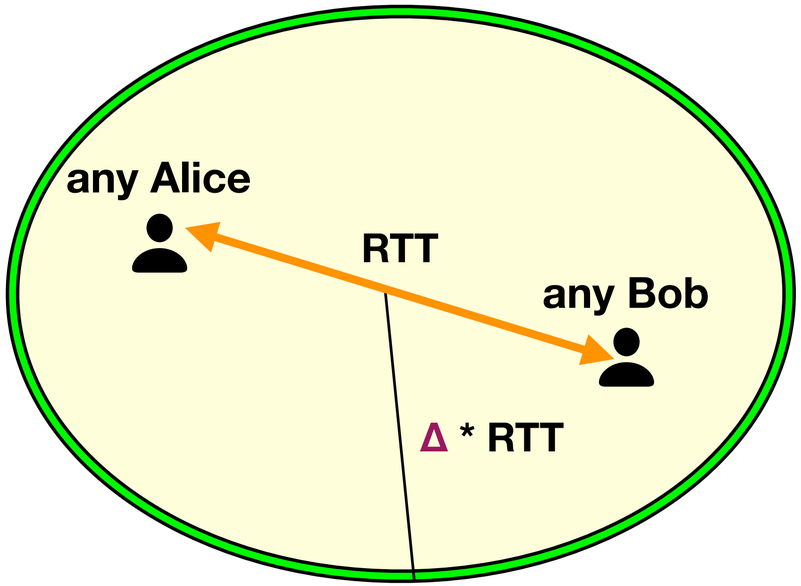
- Builds exponentially-increasing zones around each validator

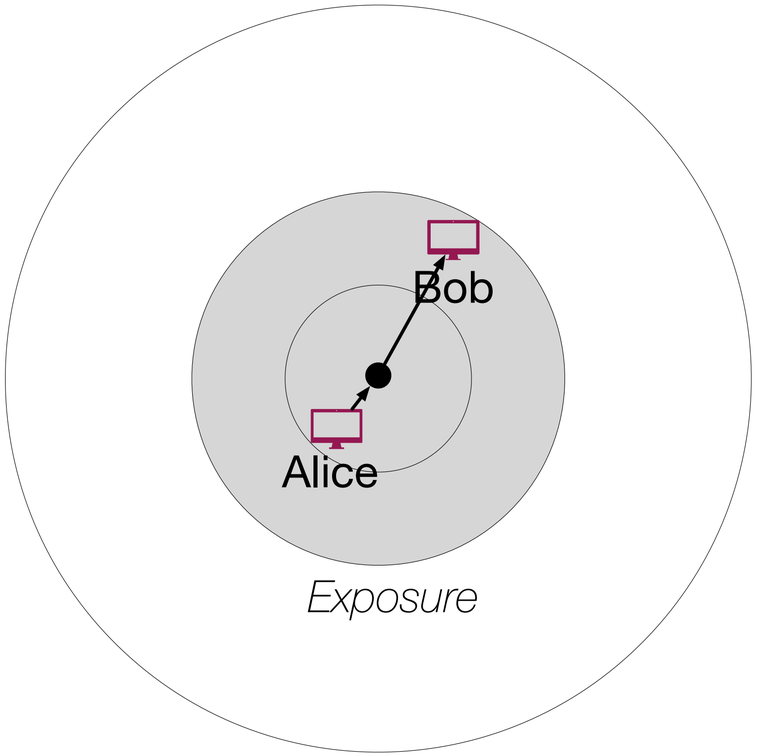




Strawman

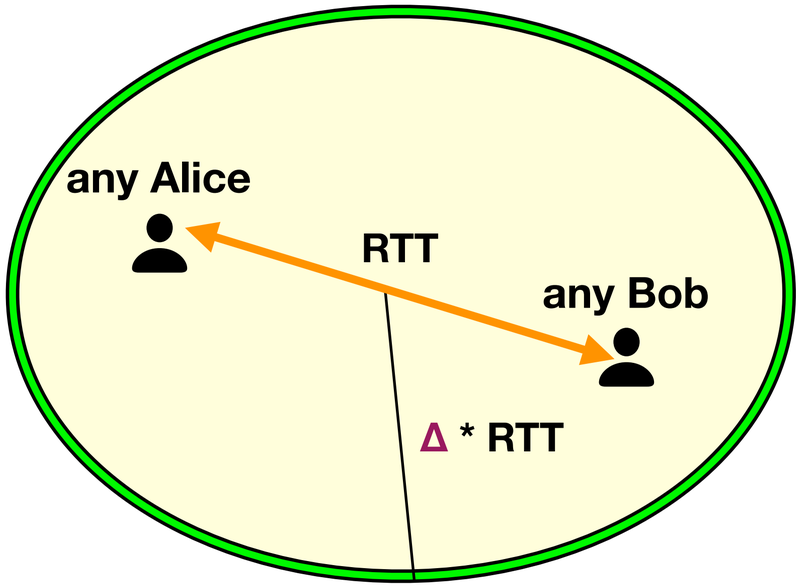
- Builds exponentially-increasing zones around each validator

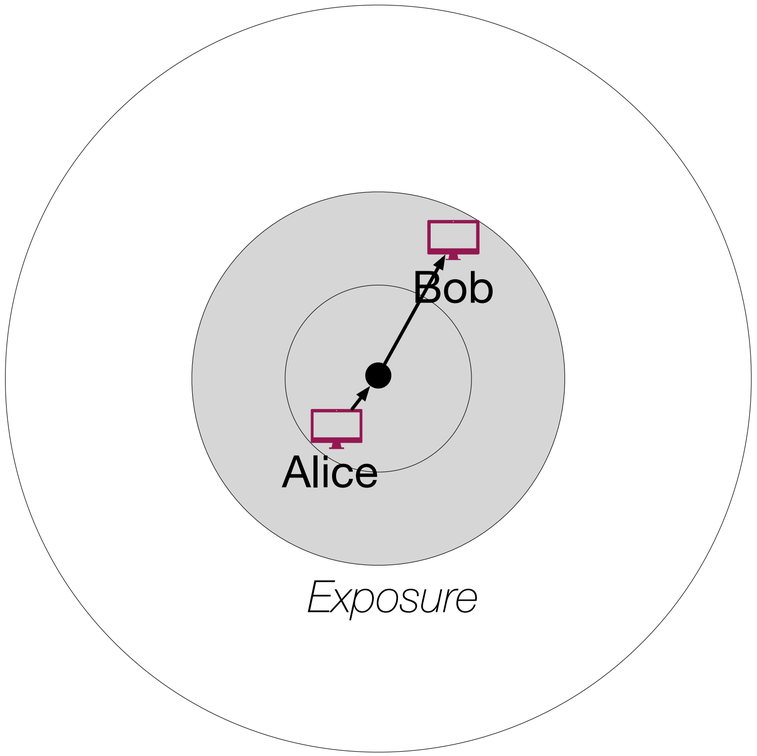




Strawman

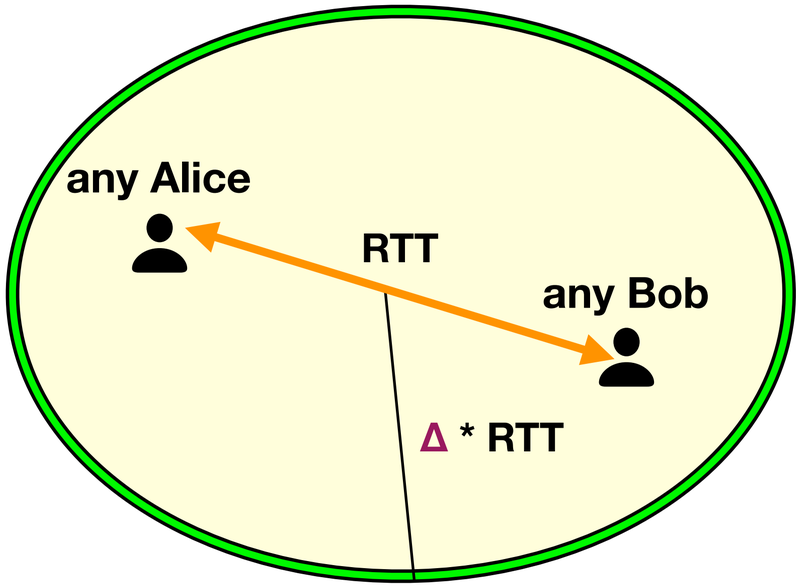
- Builds exponentially-increasing zones around each validator
  - Δ = smallest zone ≥ 2 * RTT (Alice's validator, Bob's validator)

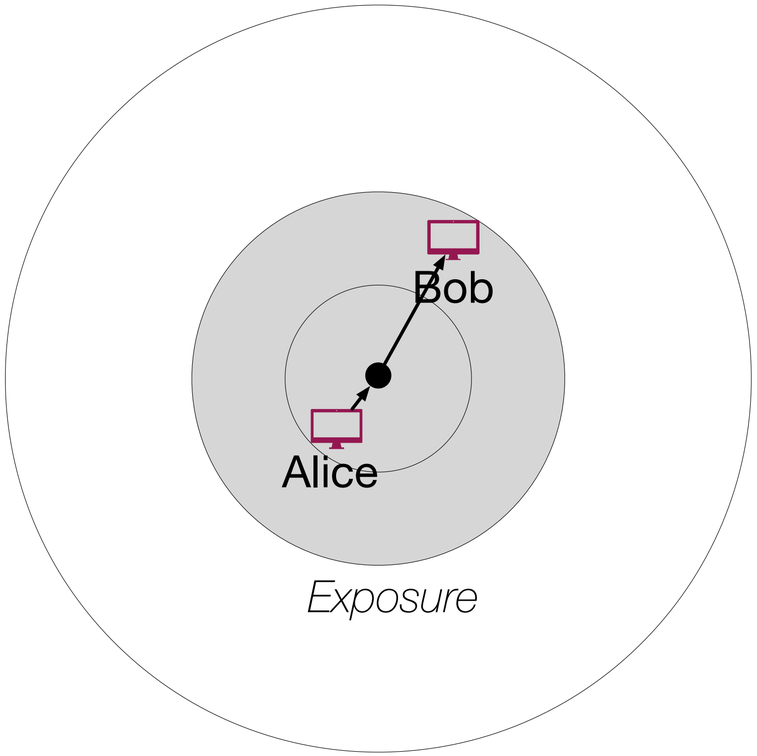




Strawman

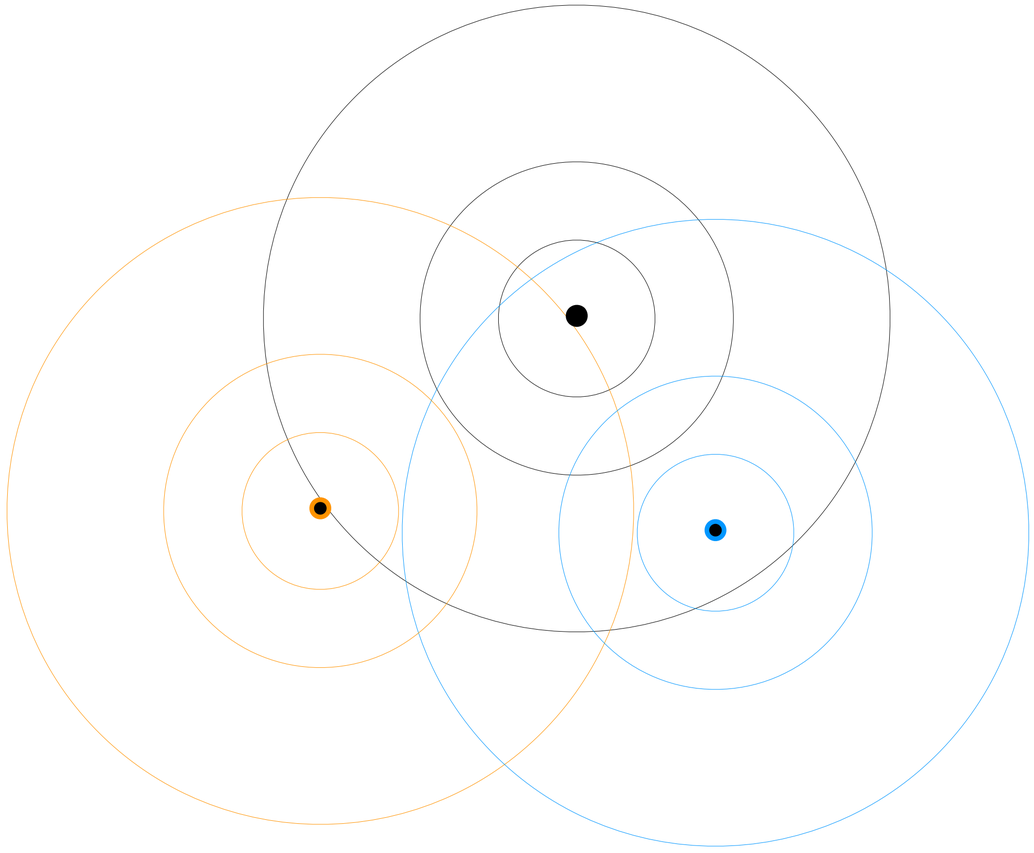
- Builds exponentially-increasing zones around each validator
  - Δ = smallest zone ≥ 2 * RTT (Alice's validator, Bob's validator)
- Overheads per validator?

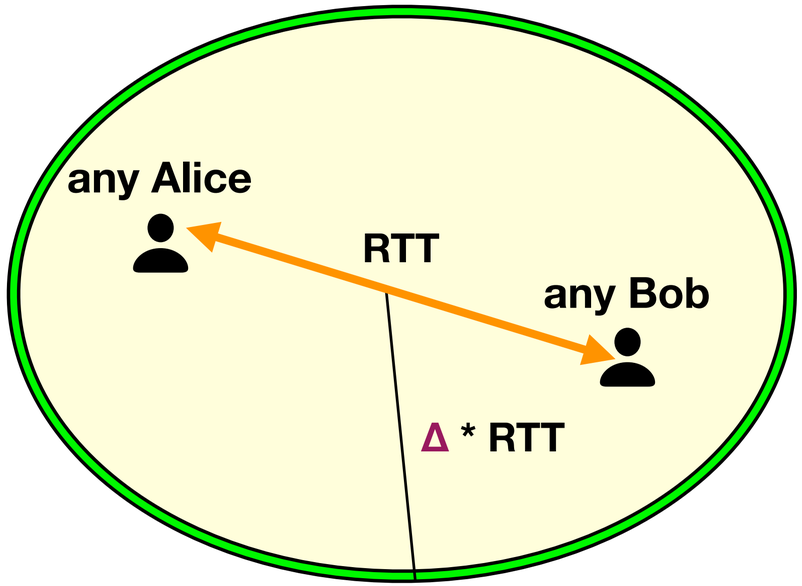




Strawman

- Builds exponentially-increasing zones around each validator
  - Δ = smallest zone ≥ 2 * RTT (Alice's validator, Bob's validator)
- Overheads per validator?
  - Each validator participates in **O(N)** zones



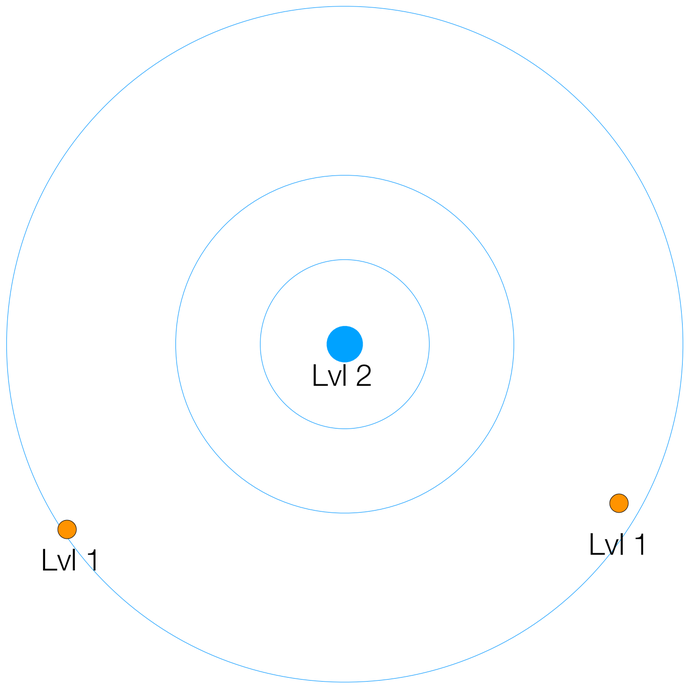


Compact metric-based zoning

- Use compact-graph approximations

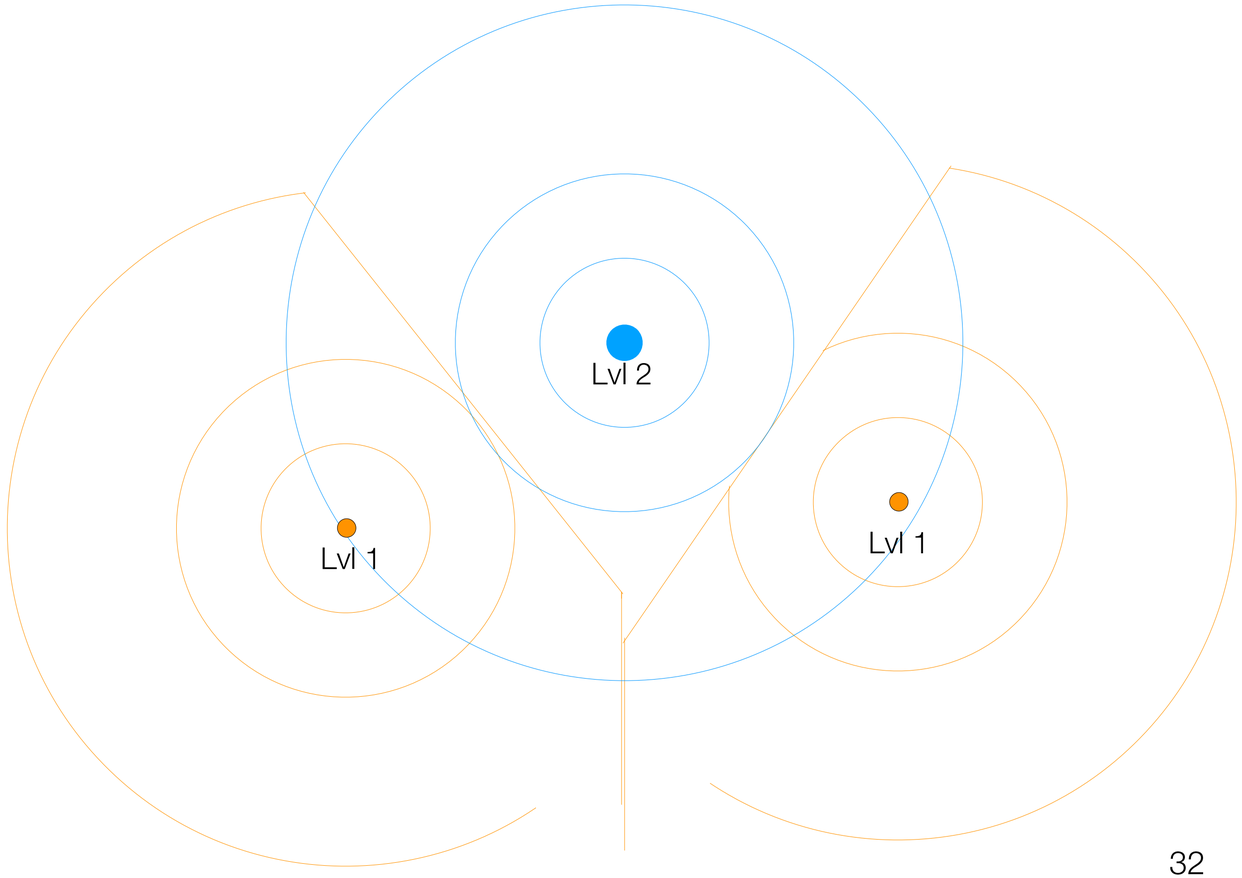
Compact metric-based zoning

- Use compact-graph approximations



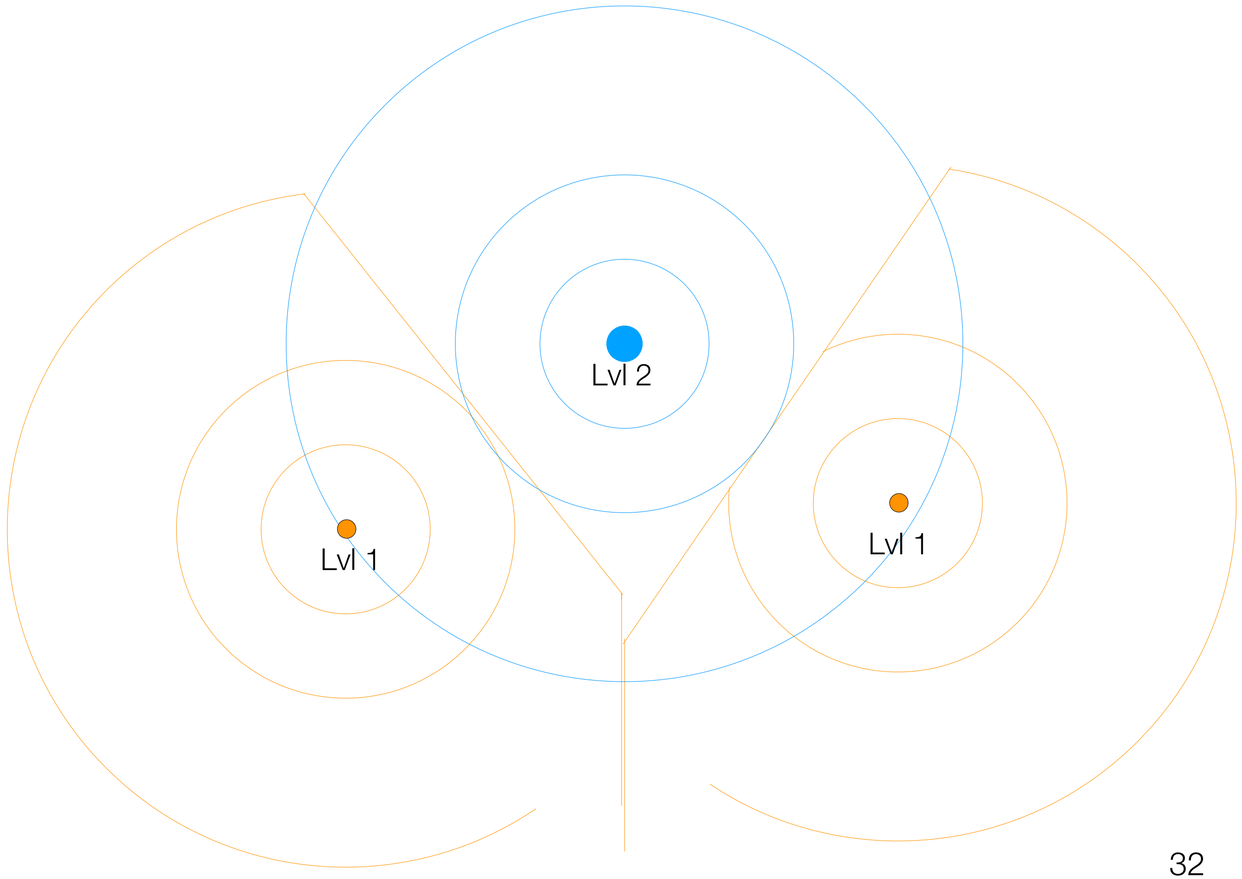
Compact metric-based zoning

- Use compact-graph approximations



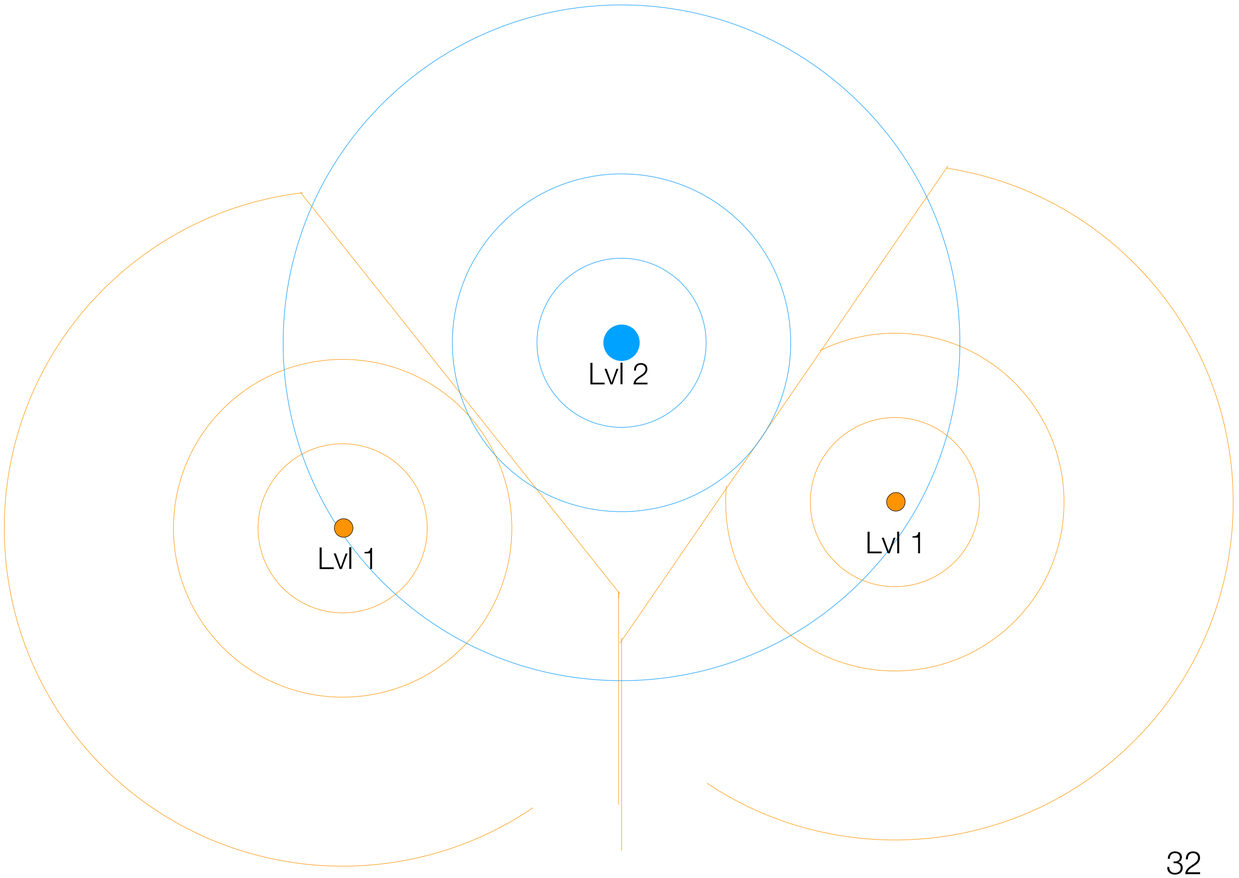
Compact metric-based zoning

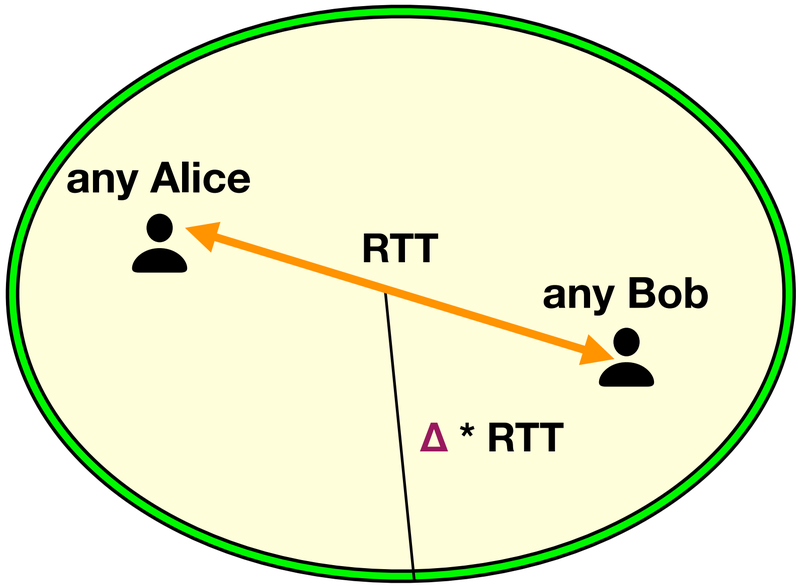
- Use compact-graph approximations
  - Each validators participates in **O(log N)** zones



Compact metric-based zoning

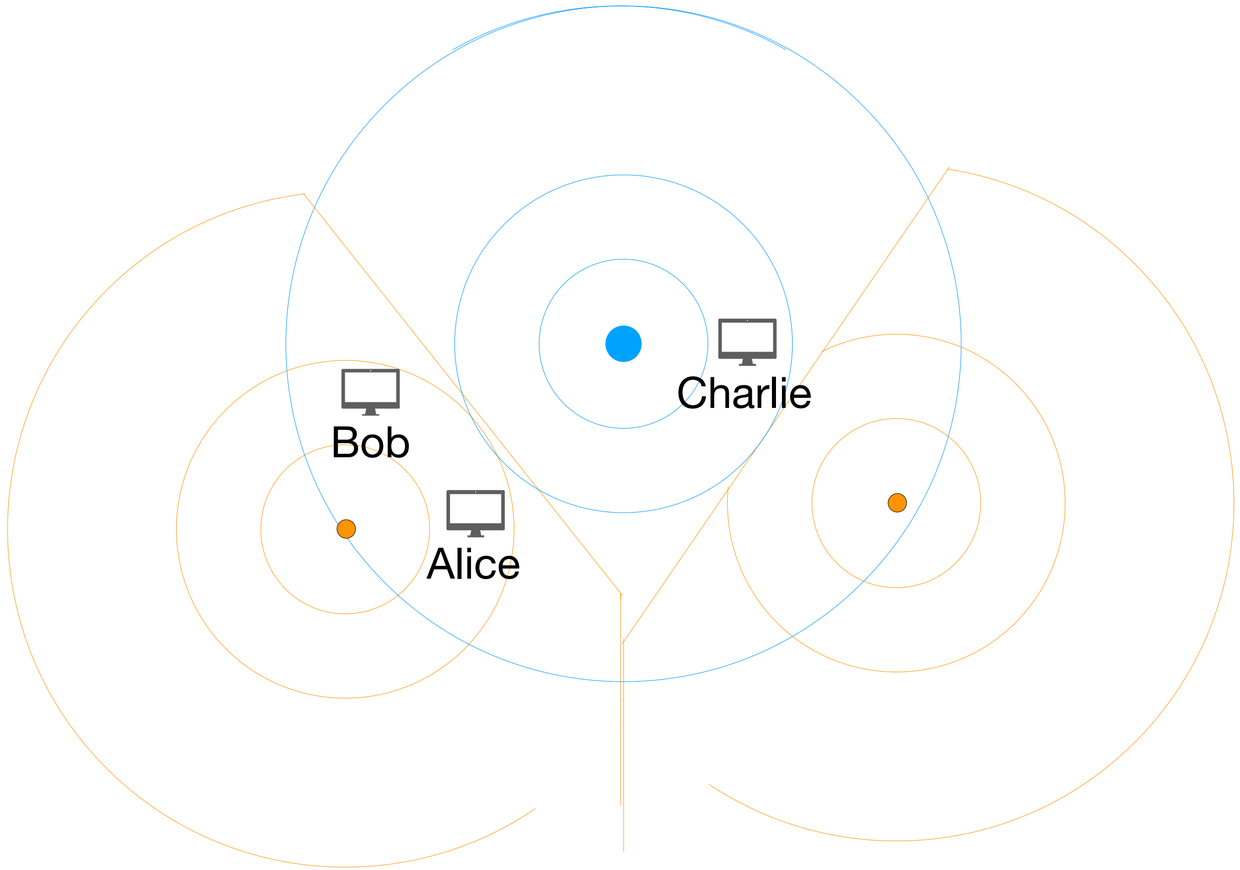
- Use compact-graph approximations
  - Each validators participates in **O(log N)** zones

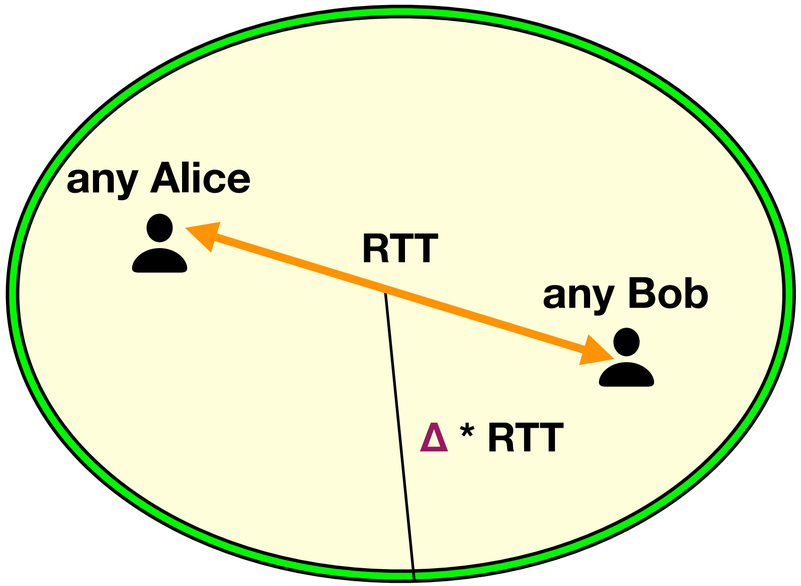




Compact metric-based zoning

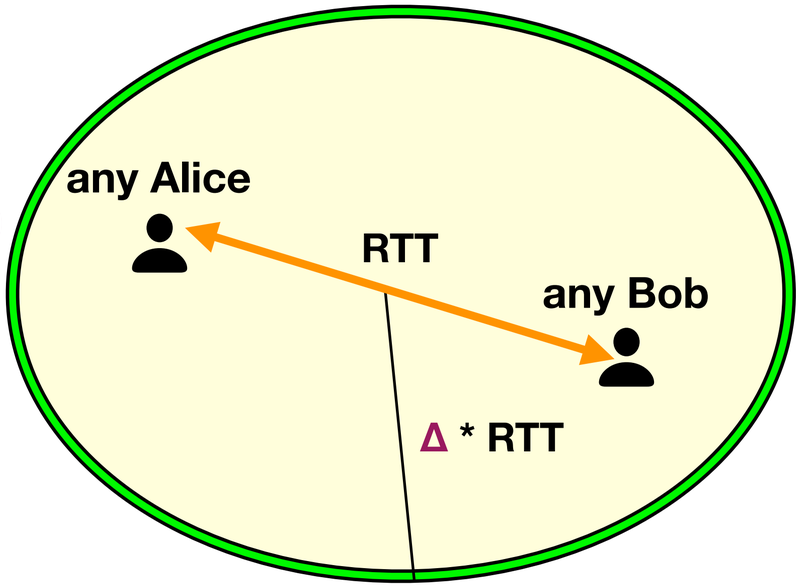
- Use compact-graph approximations
  - Each validators participates in **O(log N)** zones

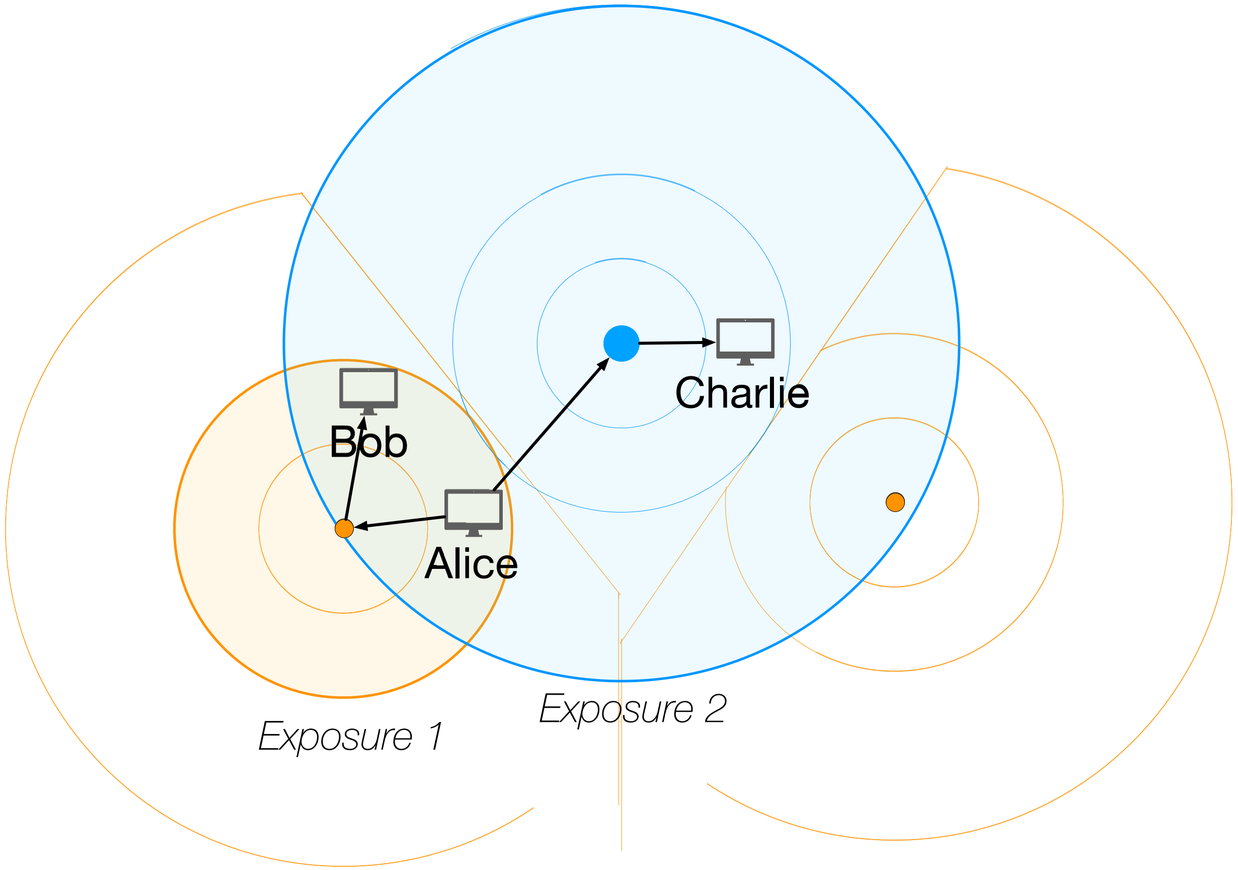




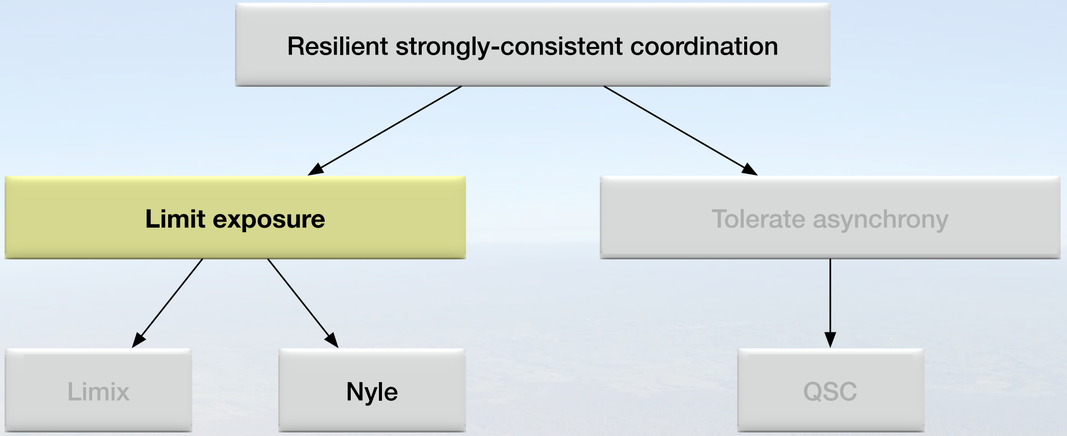
Compact metric-based zoning

- Use compact-graph approximations
  - Each validators participates in **O(log N)** zones
  - Δ = smallest zone ≥ **(2k-1)** * 2 * RTT (Alice's validator, Bob's validator)





Limiting exposure roadmap

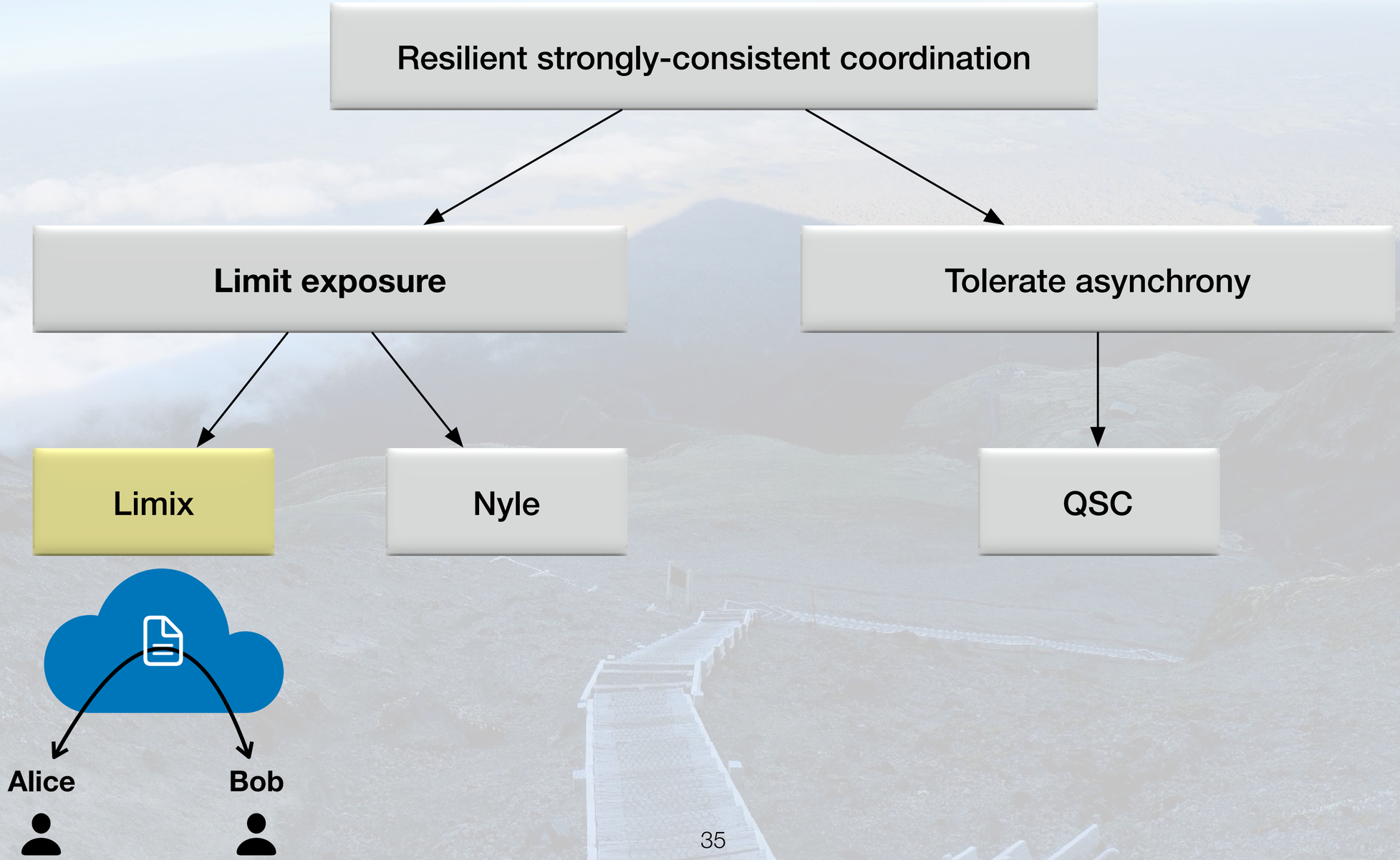


- Defining and measuring Lamport exposure
- Providing meaningful exposure guarantees
- **Summary**

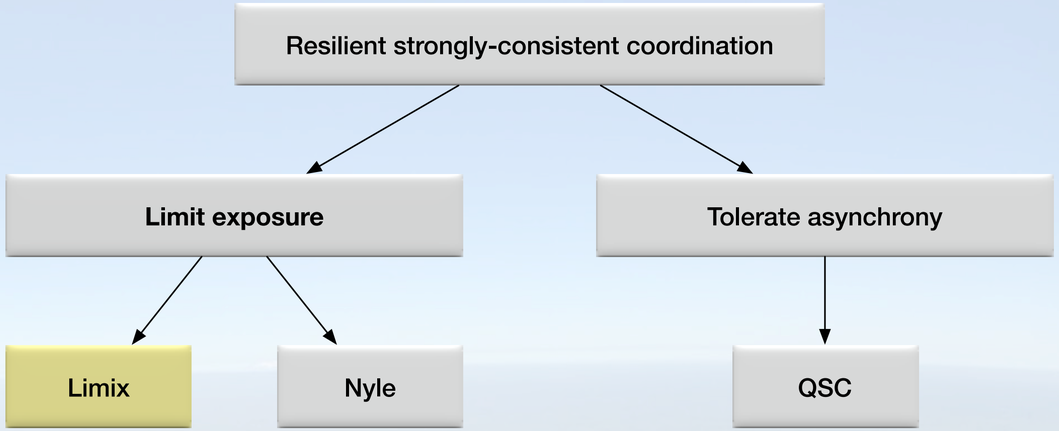
Limiting exposure summary

- **Practical metrics for assessing Lamport exposure**
- **Zoning for building globalised systems with localized resilience guarantees**

Thesis roadmap



Limix roadmap



- **The problem: exposure of metadata**
- Design of exposure-limiting metadata service
- Strongly-consistent item lookup
- Architecture
- Evaluation
- Summary

Problem: “another layer of indirection”

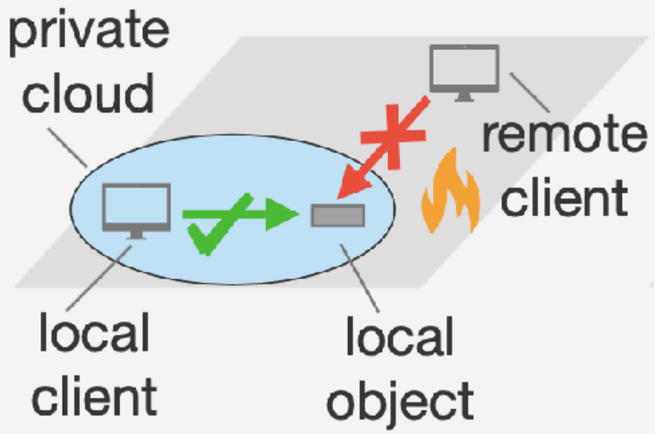
Control plane (metadata)

Data plane

Problem: “another layer of indirection”

Control plane (metadata)

Data plane



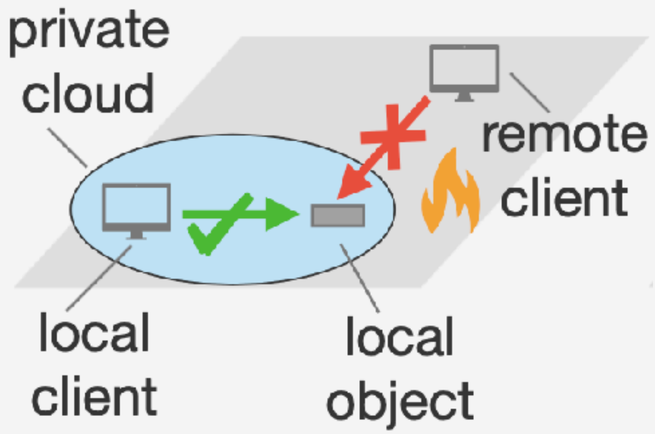
(a) local access to local infrastructure

Embed locally-scoped location in object

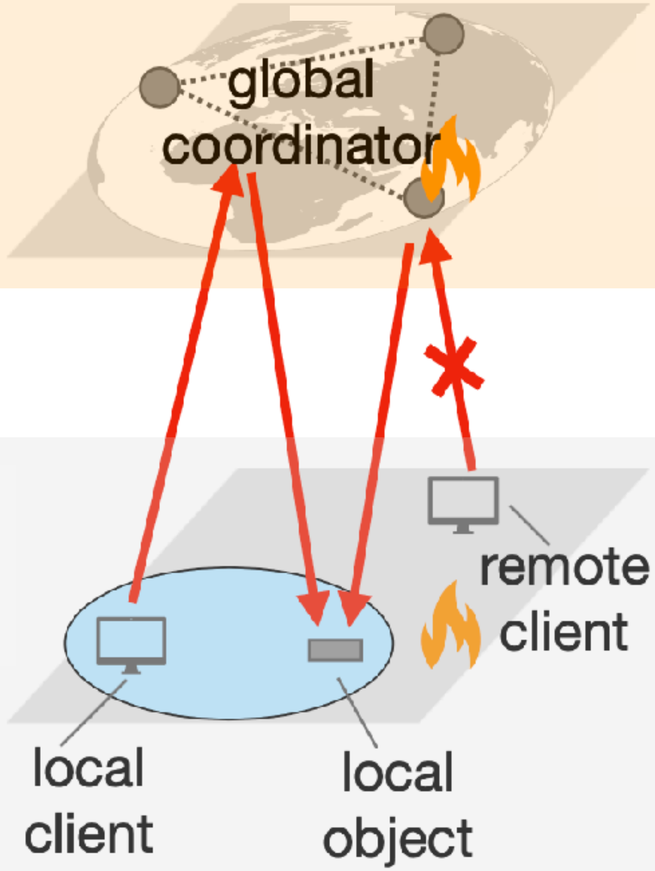
Problem: “another layer of indirection”

Control plane (metadata)

Data plane



(a) local access to local infrastructure



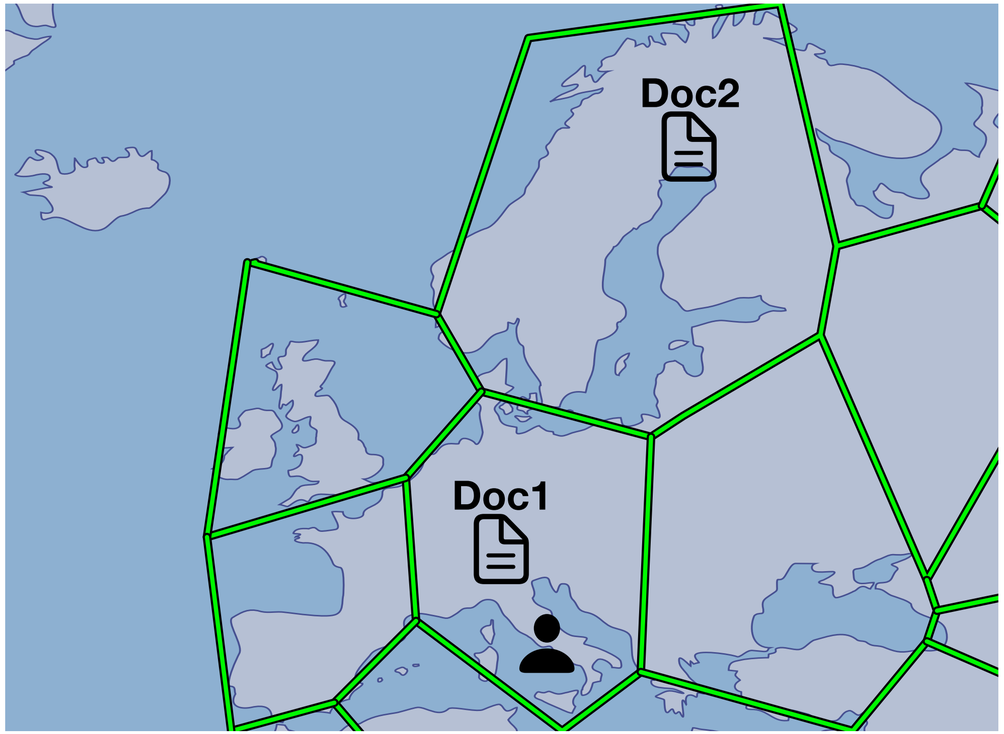
(b) access via global coordination

Embed locally-scoped location in object

Add a level of indirection

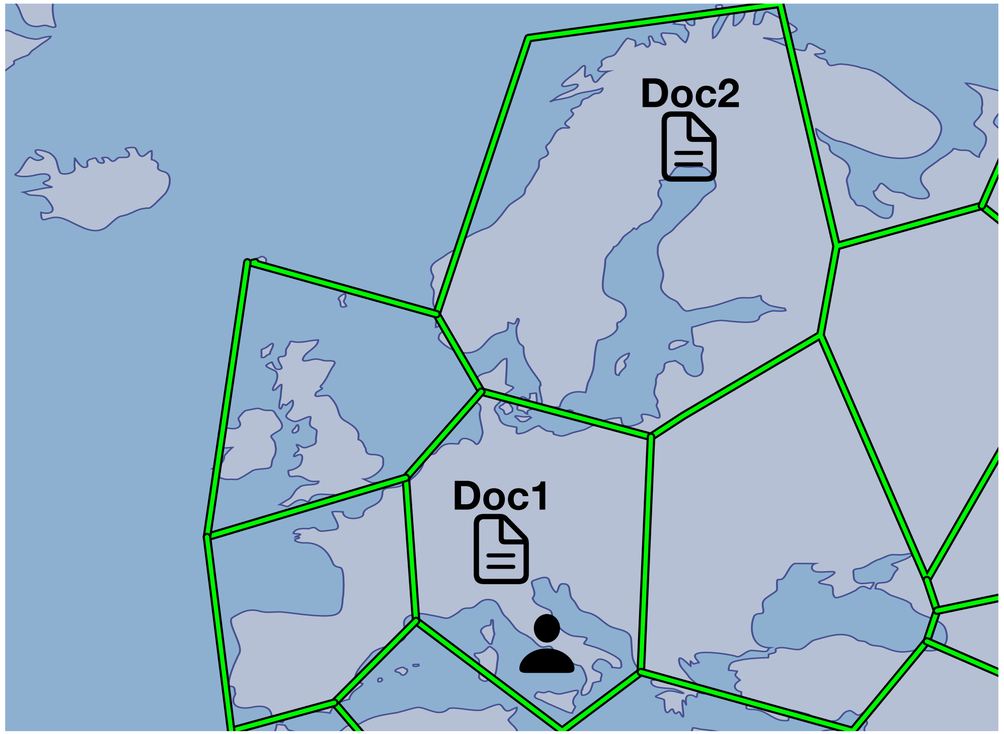
Geo-replication is not enough

Two documents in a geo-replicated system
(replicas not depicted)



Geo-replication is not enough

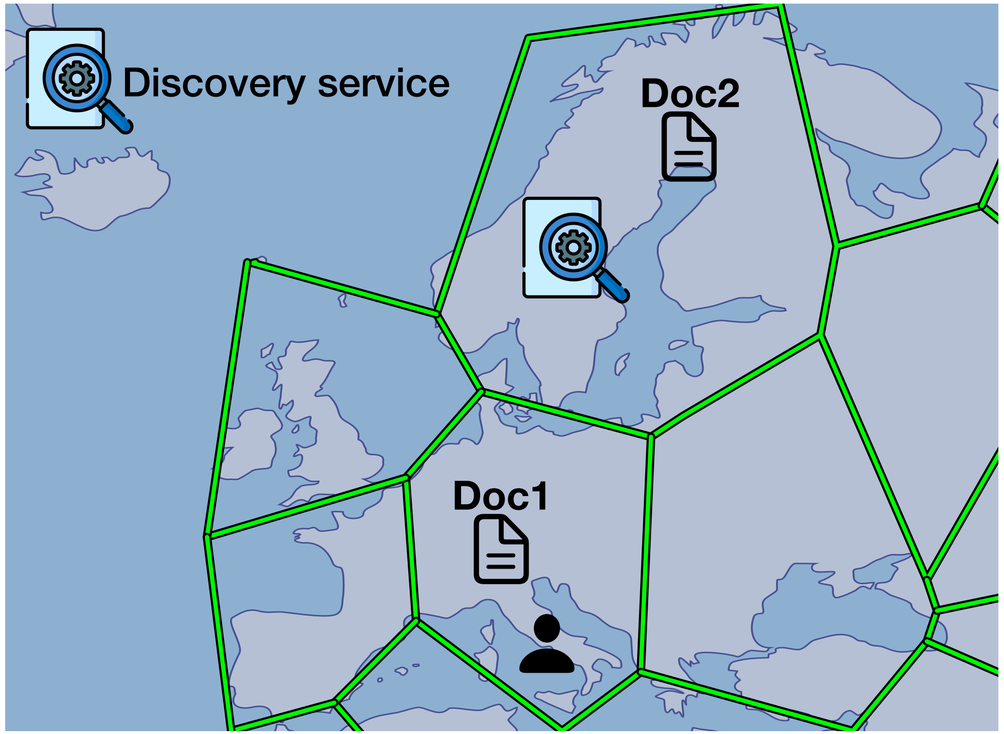
Two documents in a geo-replicated system
(replicas not depicted)



Challenge 1:
Maintain Lamport exposure while performing file lookup

Geo-replication is not enough

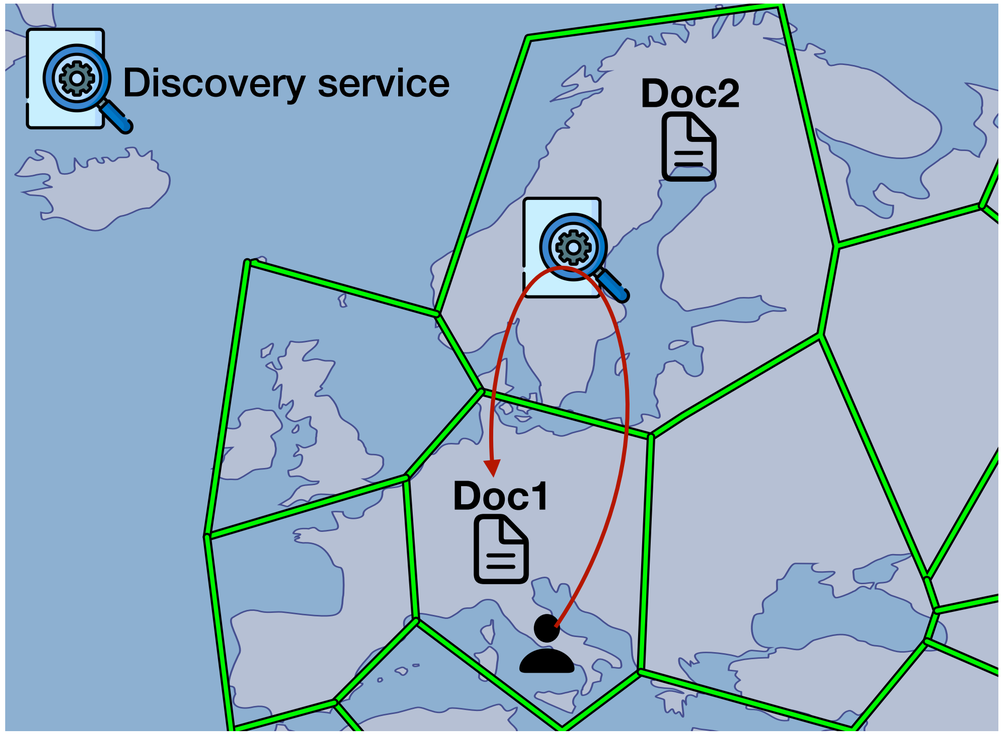
Two documents in a geo-replicated system
(replicas not depicted)



Challenge 1:
Maintain Lamport exposure while performing file lookup

Geo-replication is not enough

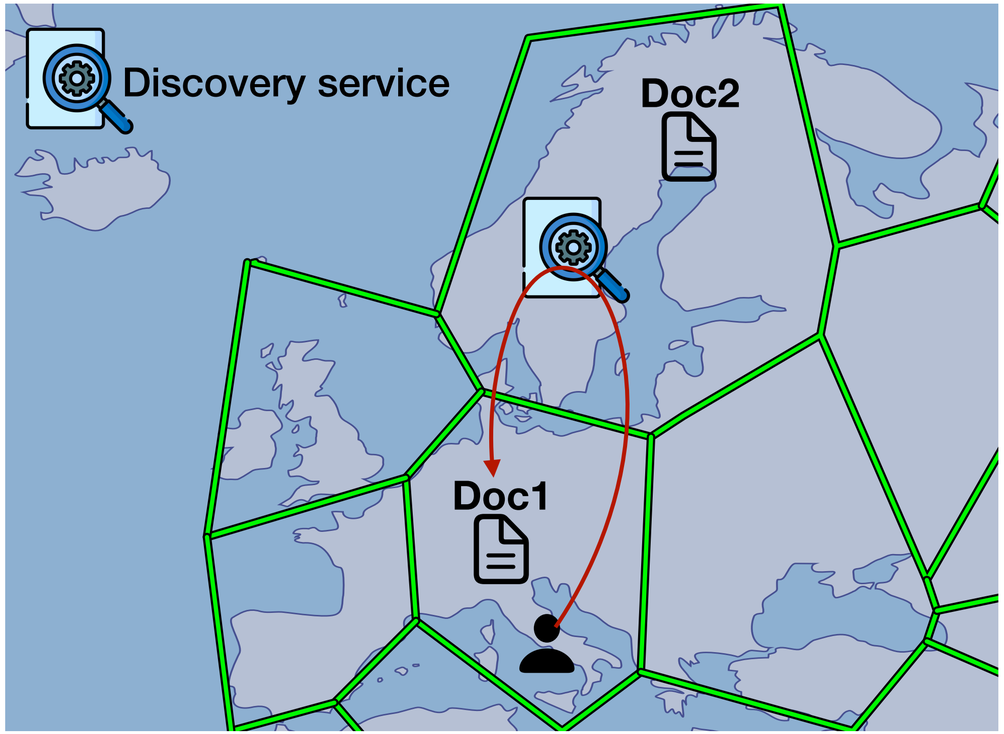
Two documents in a geo-replicated system
(replicas not depicted)



Challenge 1:
Maintain Lamport exposure while performing file lookup

Geo-replication is not enough

Two documents in a geo-replicated system
(replicas not depicted)



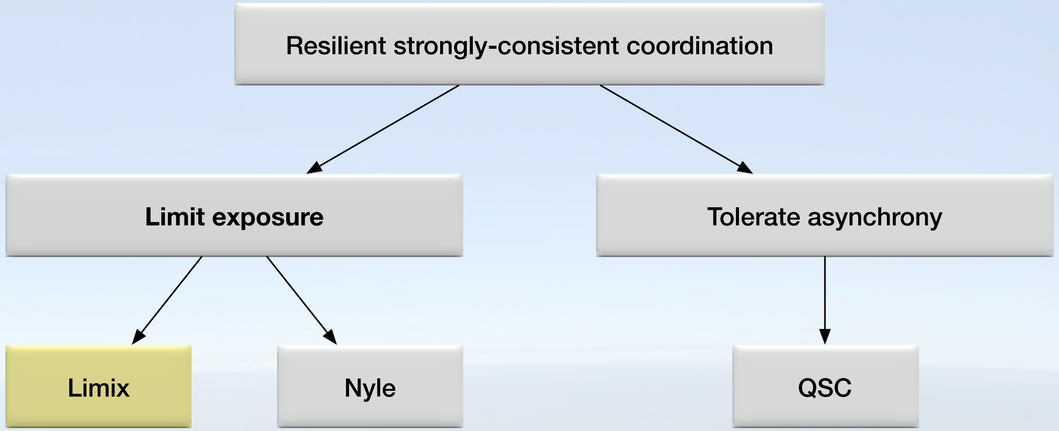
X Challenge 1:
Maintain Lamport exposure while performing file lookup

Limix focuses on the metadata problem:

How to find a localized strongly-consistent item when the location metadata is normally globalized?

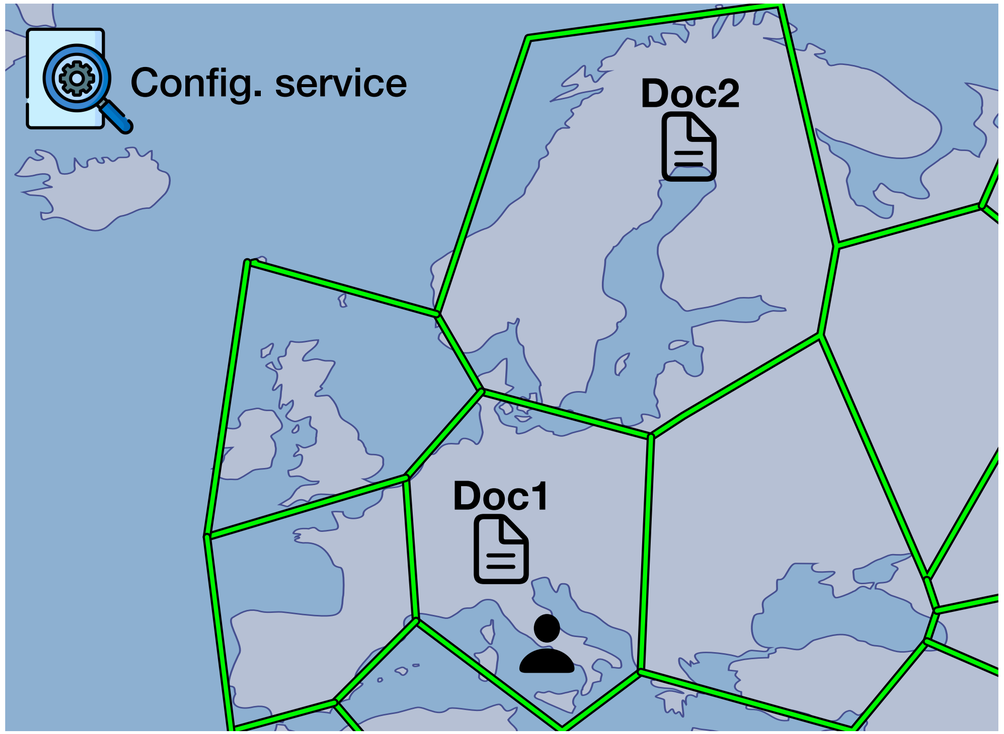
Limix roadmap

- The problem: metadata exposure
- **Design of exposure-limiting metadata service**
  - Limiting exposure of data lookup
  - Scalable metadata service
  - Fine-grained exposure
- Strongly-consistent item lookup
- Architecture
- Evaluation
- Summary



Strawman 1: limits exposure but does not scale

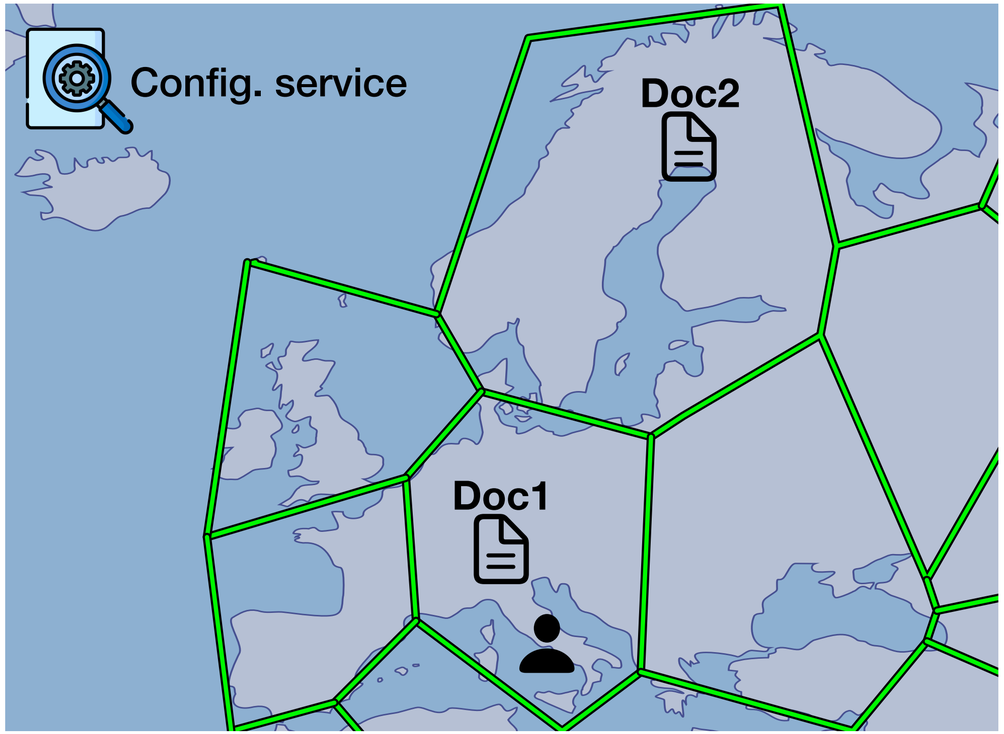
Two documents in a geo-replicated system
(replicas not depicted)



Challenge 1:
Maintain Lamport exposure while performing file lookup

Strawman 1: limits exposure but does not scale

Two documents in a geo-replicated system
(replicas not depicted)

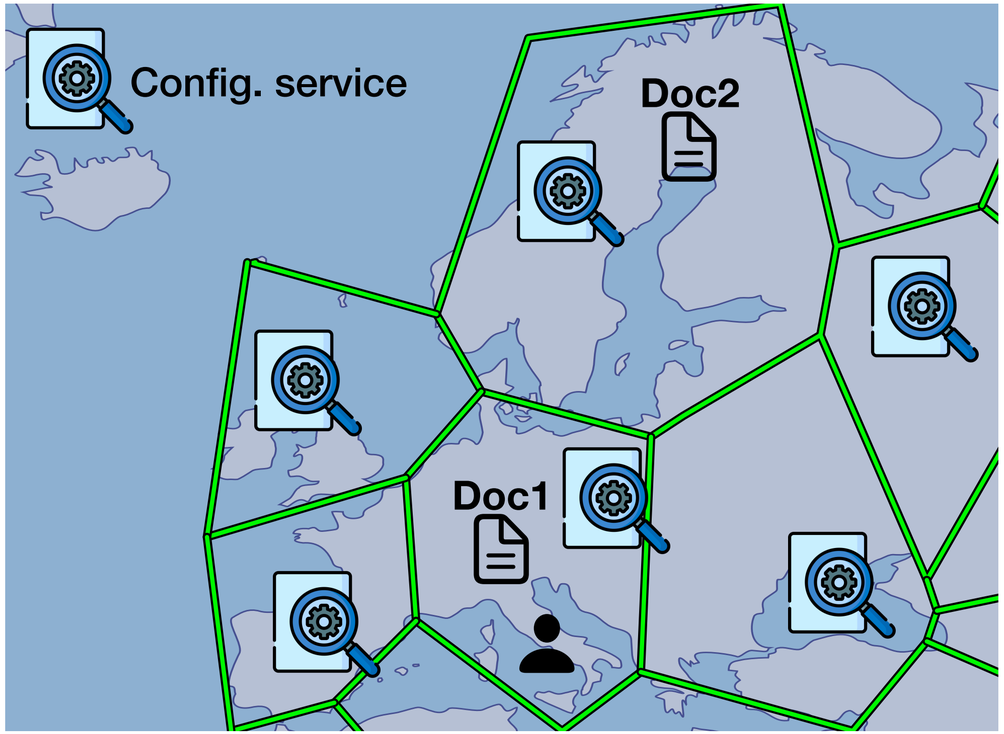


Challenge 1:
Maintain Lamport exposure while performing file lookup

Solution:
Replicate the discovery service

Strawman 1: limits exposure but does not scale

Two documents in a geo-replicated system
(replicas not depicted)

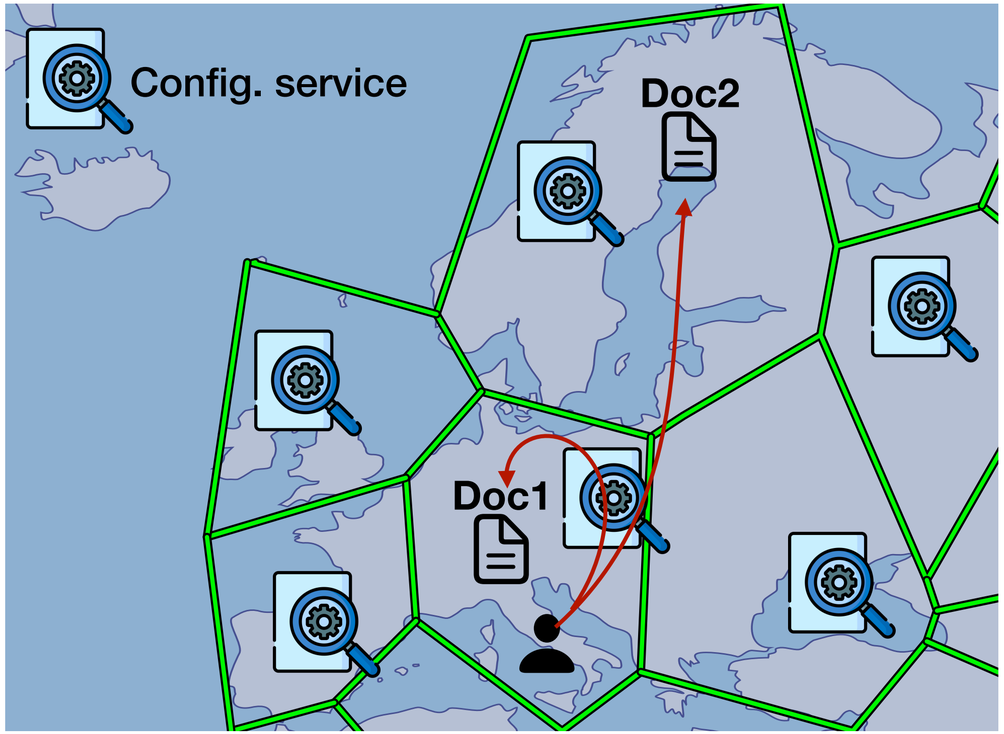


Challenge 1:
Maintain Lamport exposure while performing file lookup

Solution:
Replicate the discovery service

Strawman 1: limits exposure but does not scale

Two documents in a geo-replicated system
(replicas not depicted)

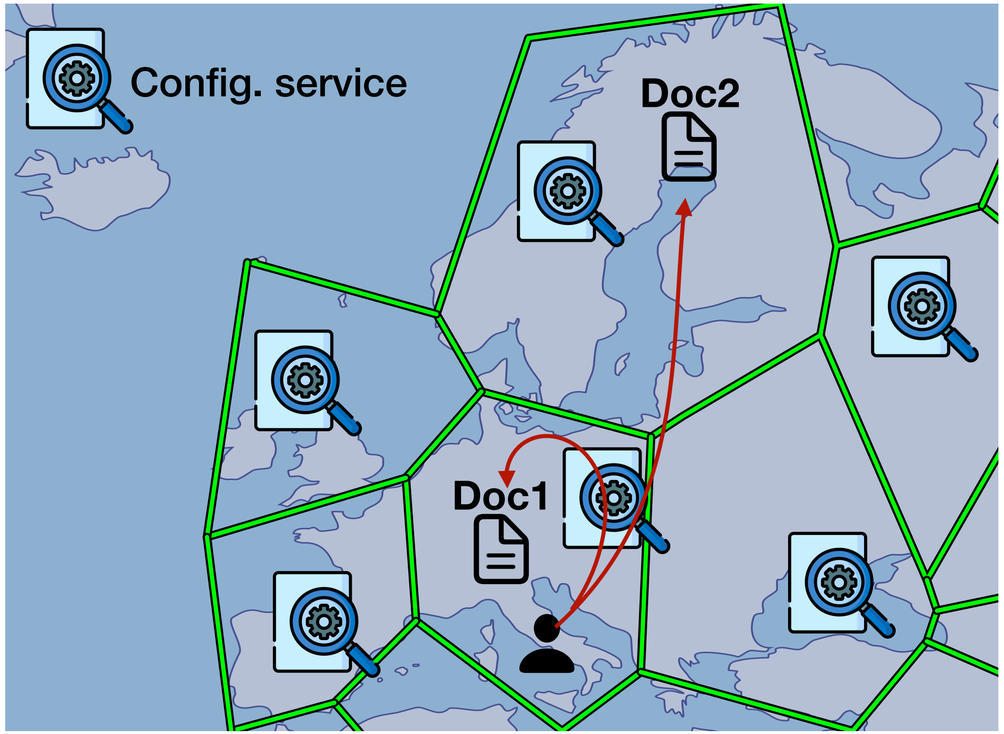


Challenge 1:
Maintain Lamport exposure while performing file lookup

Solution:
Replicate the discovery service

Strawman 1: limits exposure but does not scale

Two documents in a geo-replicated system
(replicas not depicted)

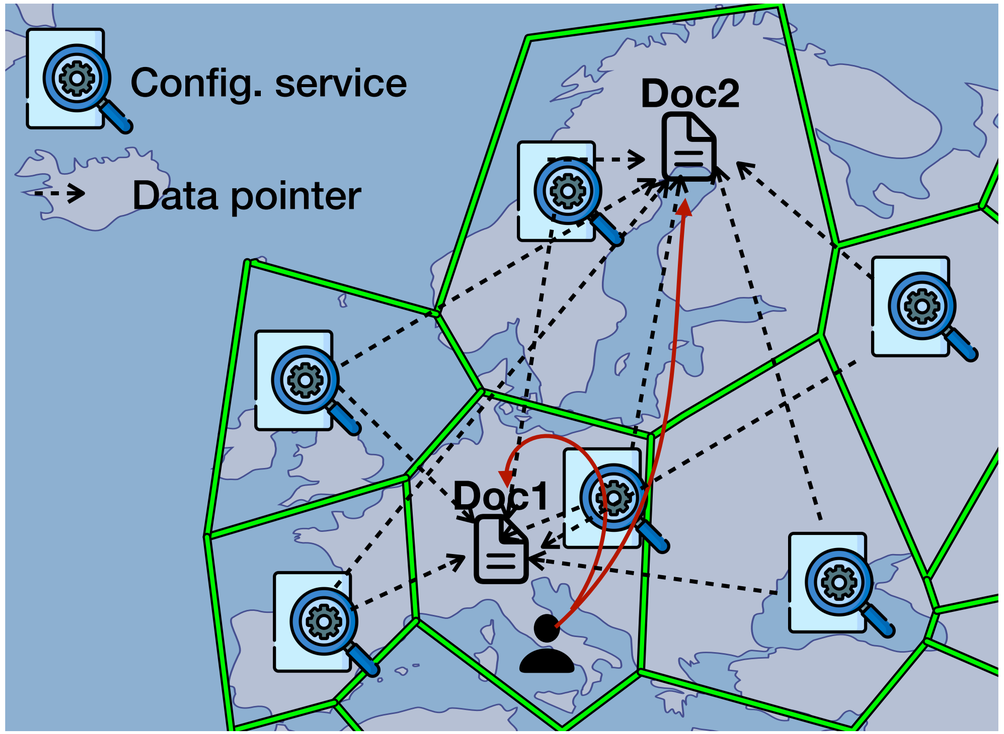


√ **Challenge 1:**
Maintain Lamport exposure while performing file lookup

Solution:
Replicate the discovery service

Strawman 1: limits exposure but does not scale

Two documents in a geo-replicated system
(replicas not depicted)

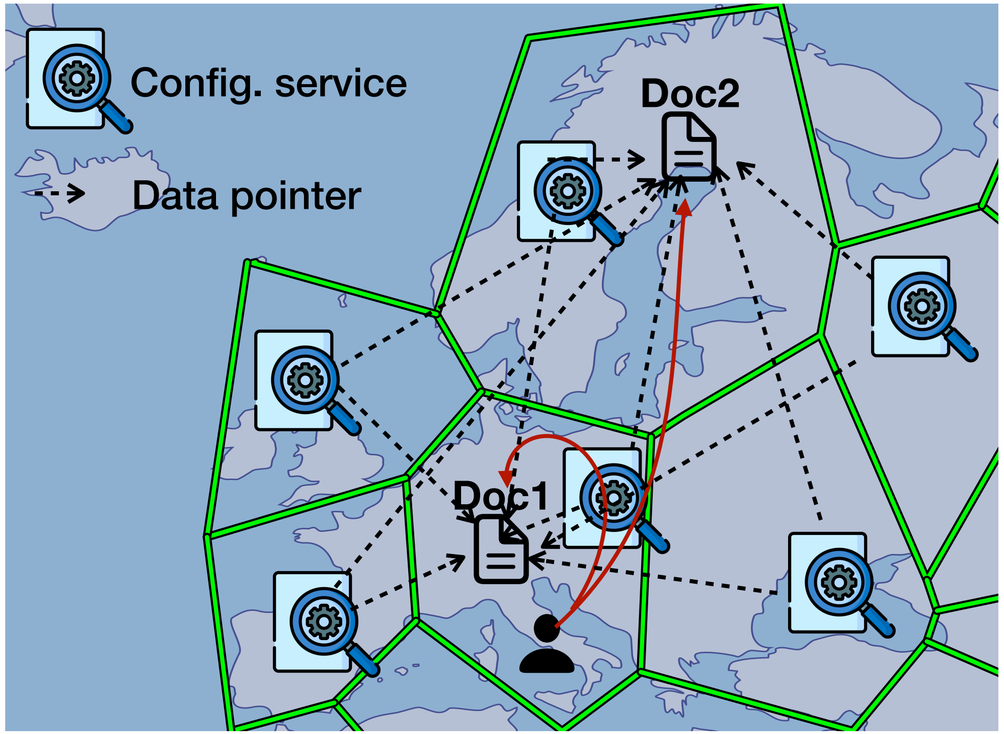


√ **Challenge 1:**
Maintain Lamport exposure while performing file lookup

Solution:
Replicate the discovery service
Per-zone discovery service, each service points to all data

Strawman 1: limits exposure but does not scale

Two documents in a geo-replicated system
(replicas not depicted)



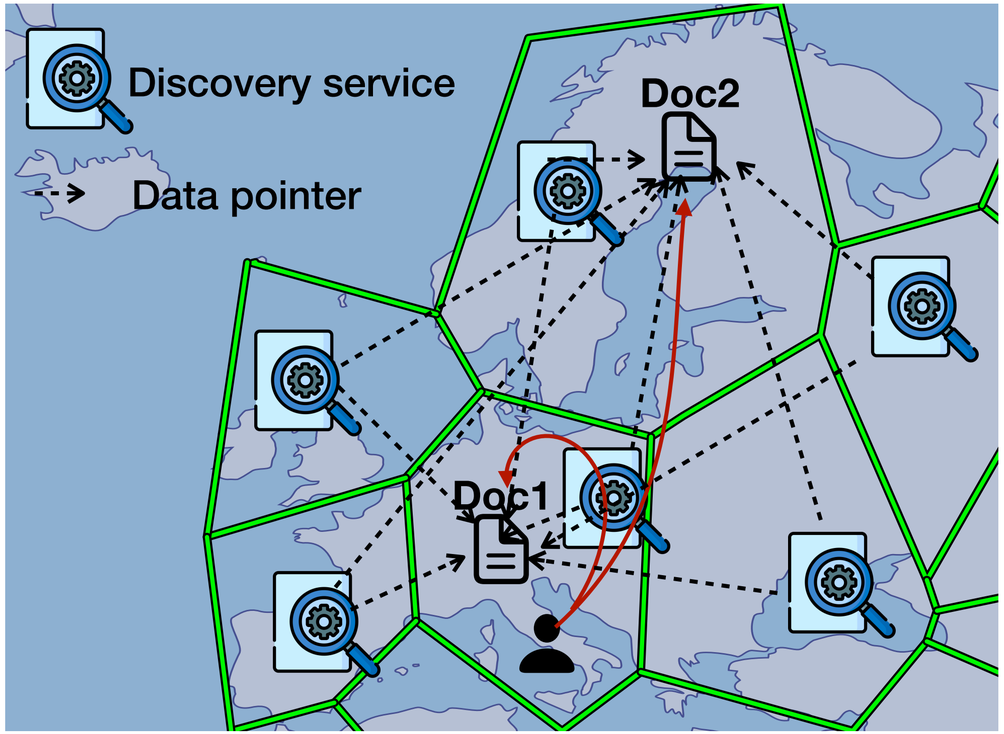
√ **Challenge 1:**
Maintain Lamport exposure while performing file lookup

Solution:
Replicate the discovery service
Per-zone discovery service, each service points to all data

X **Challenge 2:**
Scalable discovery service enforcing Lamport exposure

Strawman 2: scales but provides coarse-grained exposure

Two documents in a geo-replicated system
(replicas not depicted)



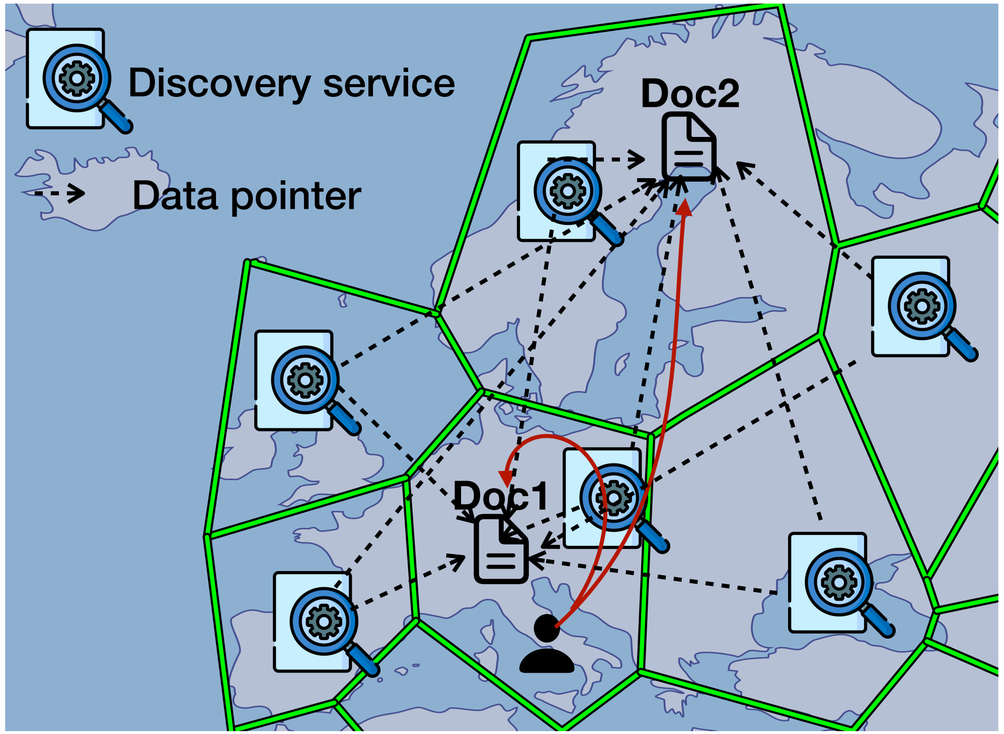
√ **Challenge 1:**
Maintain Lamport exposure while performing file lookup

Solution:
Replicate the discovery service

Challenge 2:
Scalable discovery service enforcing Lamport exposure

Strawman 2: scales but provides coarse-grained exposure

Two documents in a geo-replicated system
(replicas not depicted)



√ **Challenge 1:**
Maintain Lamport exposure while performing file lookup

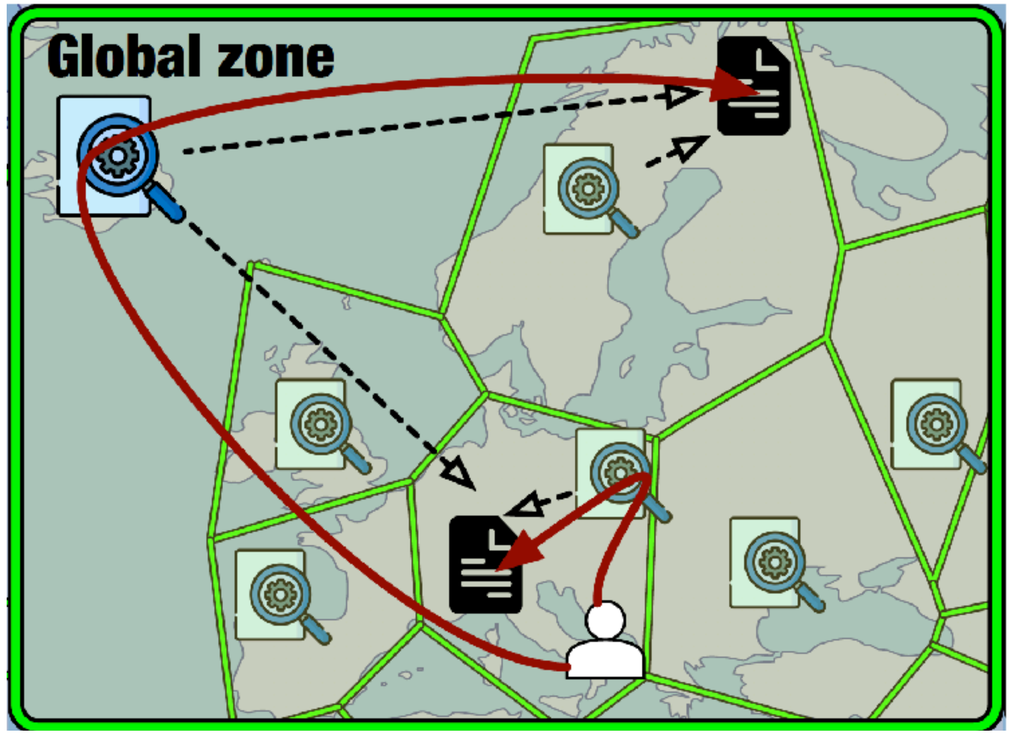
Solution:
Replicate the discovery service

Challenge 2:
Scalable discovery service enforcing Lamport exposure

Solution:
Add a global zone
The per-zone discovery service points to local data only

Strawman 2: scales but provides coarse-grained exposure

Two documents in a geo-replicated system
(replicas not depicted)



√ **Challenge 1:**
Maintain Lamport exposure while performing file lookup

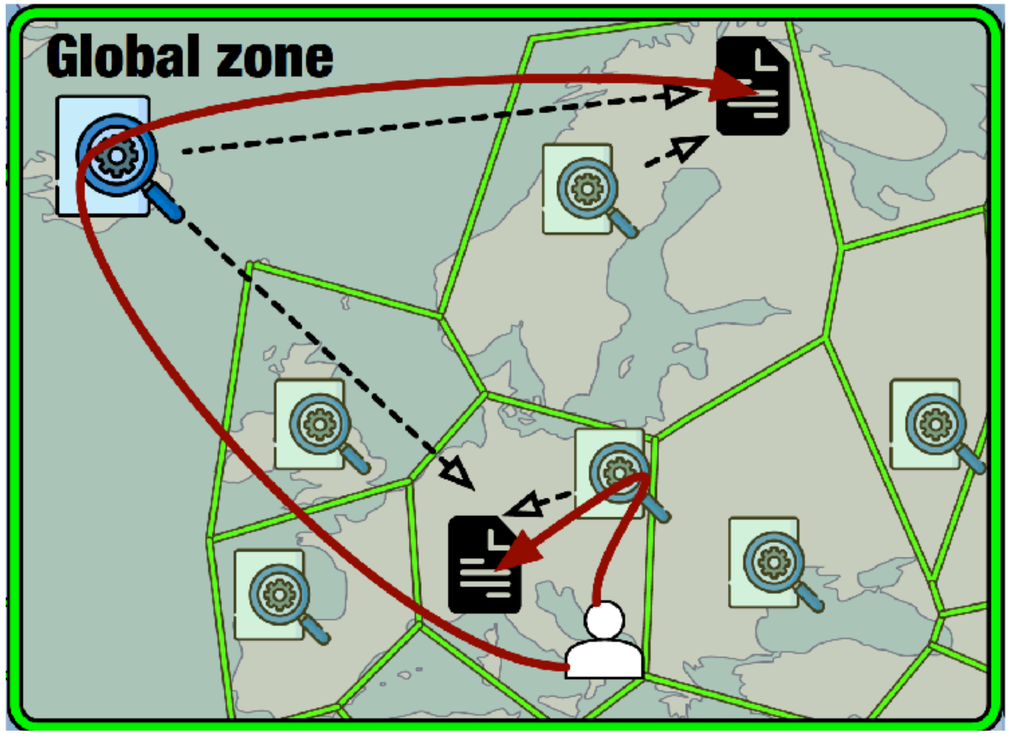
Solution:
Replicate the discovery service

Challenge 2:
Scalable discovery service enforcing Lamport exposure

Solution:
Add a global zone
The per-zone discovery service points to local data only

Strawman 2: scales but provides coarse-grained exposure

Two documents in a geo-replicated system
(replicas not depicted)



√ **Challenge 1:**
Maintain Lamport exposure while performing file lookup

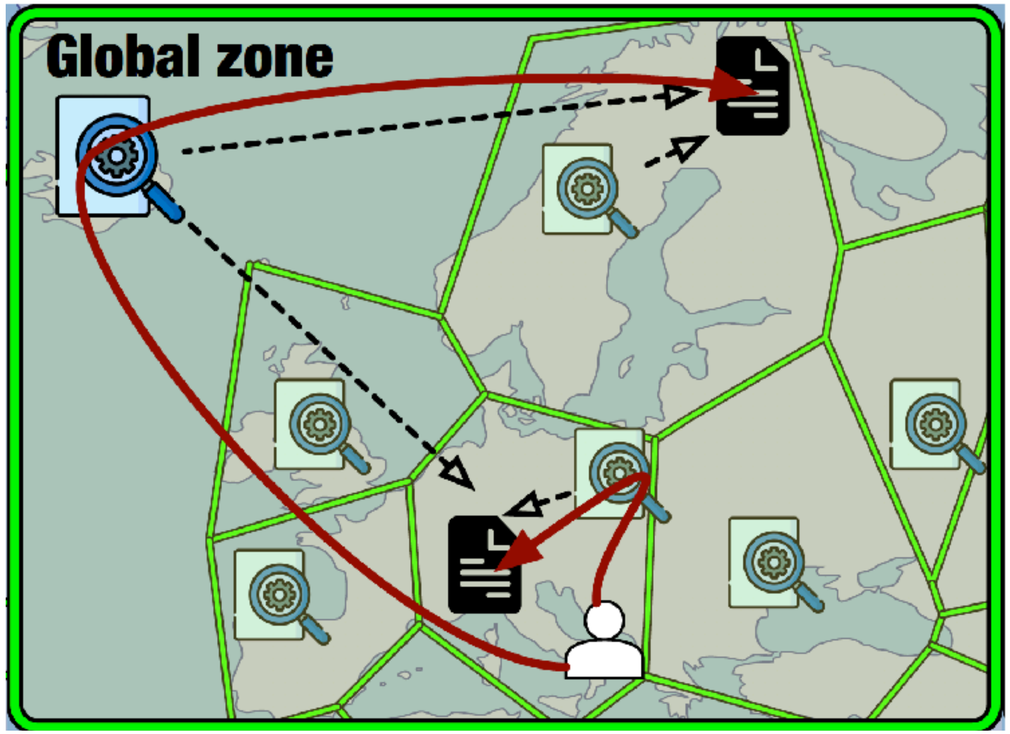
Solution:
Replicate the discovery service

√ **Challenge 2:**
Scalable discovery service enforcing Lamport exposure

Solution:
Add a global zone
The per-zone discovery service points to local data only

Strawman 2: scales but provides coarse-grained exposure

Two documents in a geo-replicated system
(replicas not depicted)



√ **Challenge 1:**
Maintain Lamport exposure while performing file lookup

Solution:
Replicate the discovery service

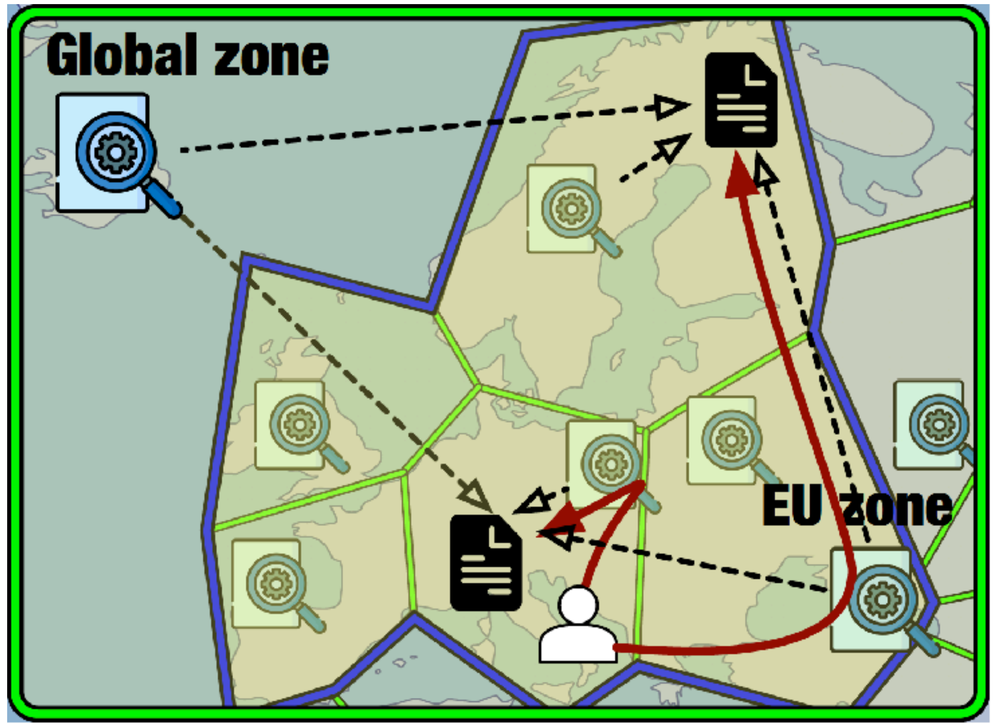
√ **Challenge 2:**
Scalable discovery service enforcing Lamport exposure

Solution:
Add a global zone
The per-zone discovery service points to local data only

X **Challenge 3**
Fine-grained overlapping Lamport exposure

Limix: our proposal for limiting Lamport exposure

Two documents in a geo-replicated system
(replicas not depicted)



√ **Challenge 1:**
Maintain Lamport exposure while performing file lookup

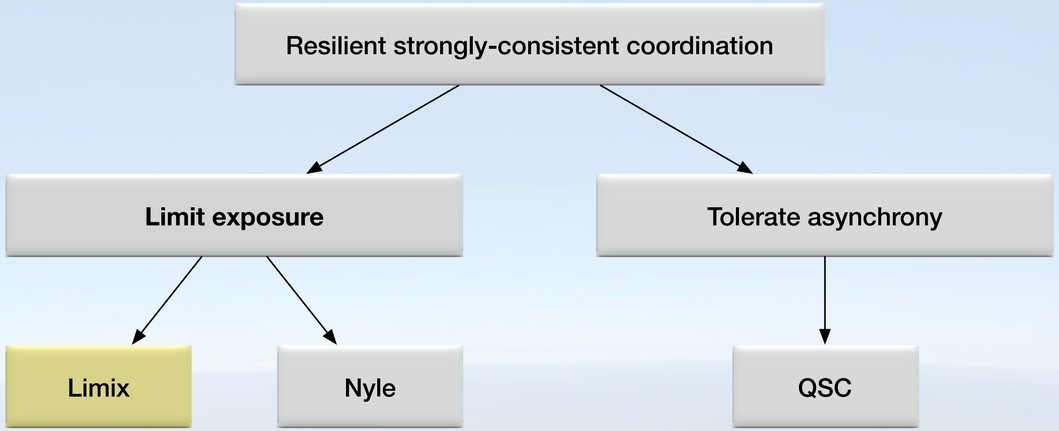
Solution:
Replicate the discovery service

√ **Challenge 2:**
Scalable discovery service enforcing Lamport exposure

Solution:
Add a global zone
The per-zone discovery service points to local data only

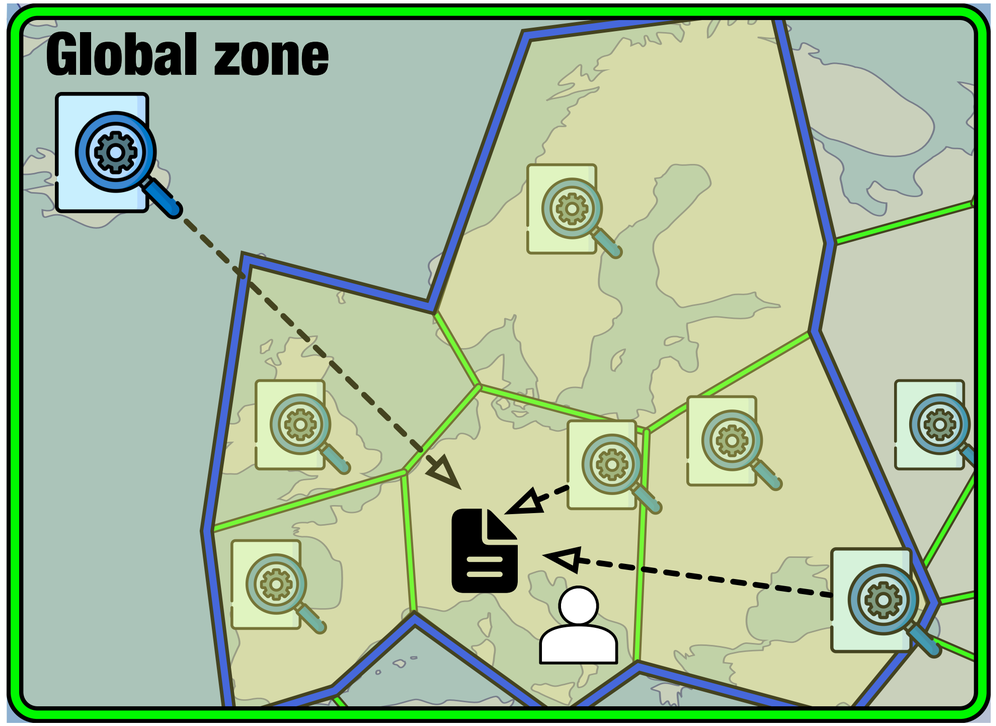
√ **Challenge 3**
Fine-grained overlapping Lamport exposure

Limix roadmap



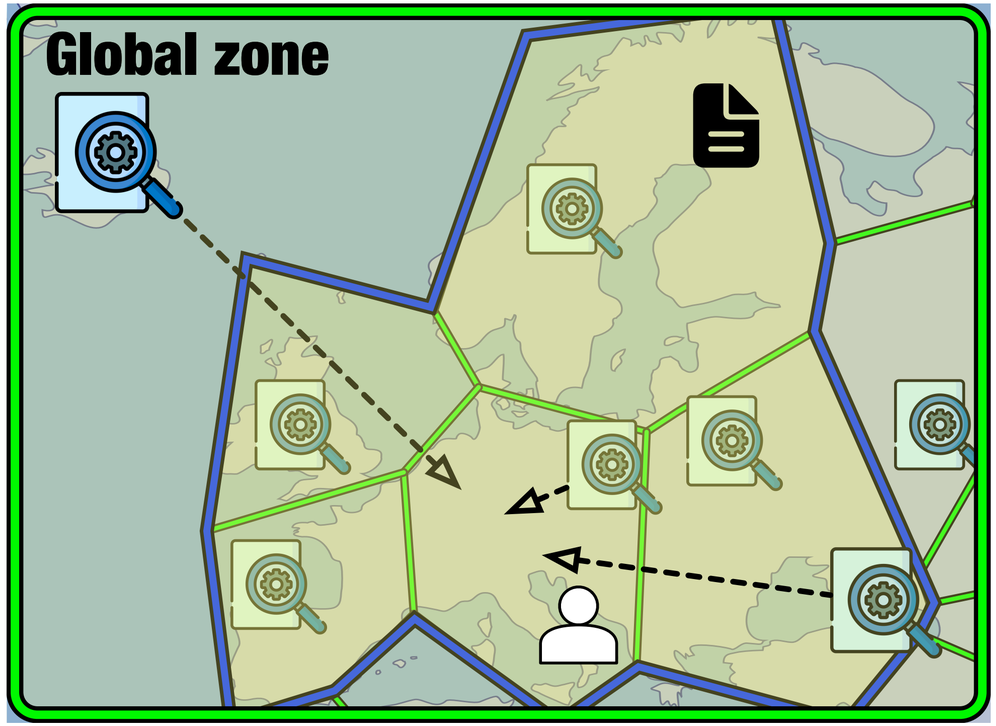
- The problem: metadata exposure
- Design of exposure-limiting metadata service
- **Strongly-consistent item lookup**
- Architecture
- Evaluation
- Summary

Locating latest data version during migration



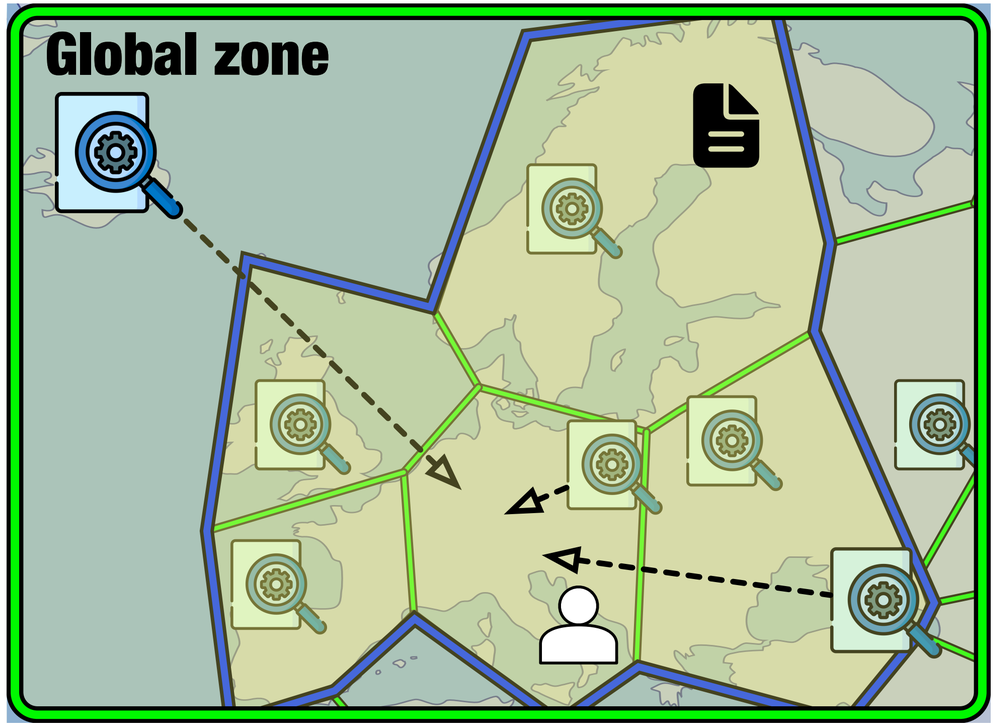
Challenge 1: Inconsistent metadata during document migration

Locating latest data version during migration



Challenge 1: Inconsistent metadata during document migration

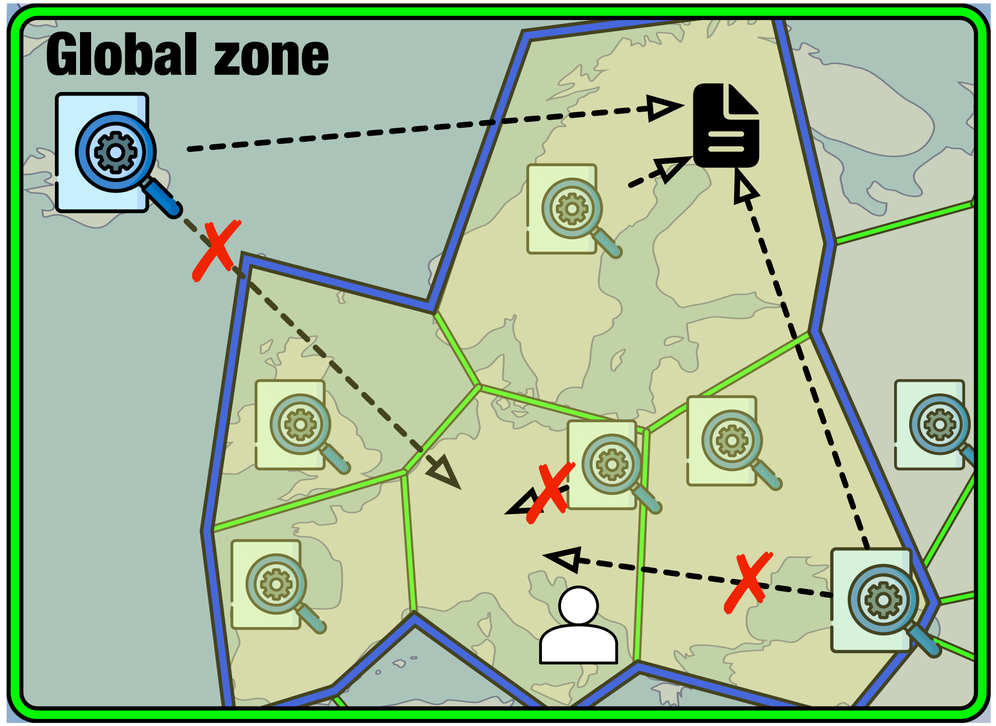
Locating latest data version during migration



Challenge 1: Inconsistent metadata during document migration

Solution: Update all relevant DS

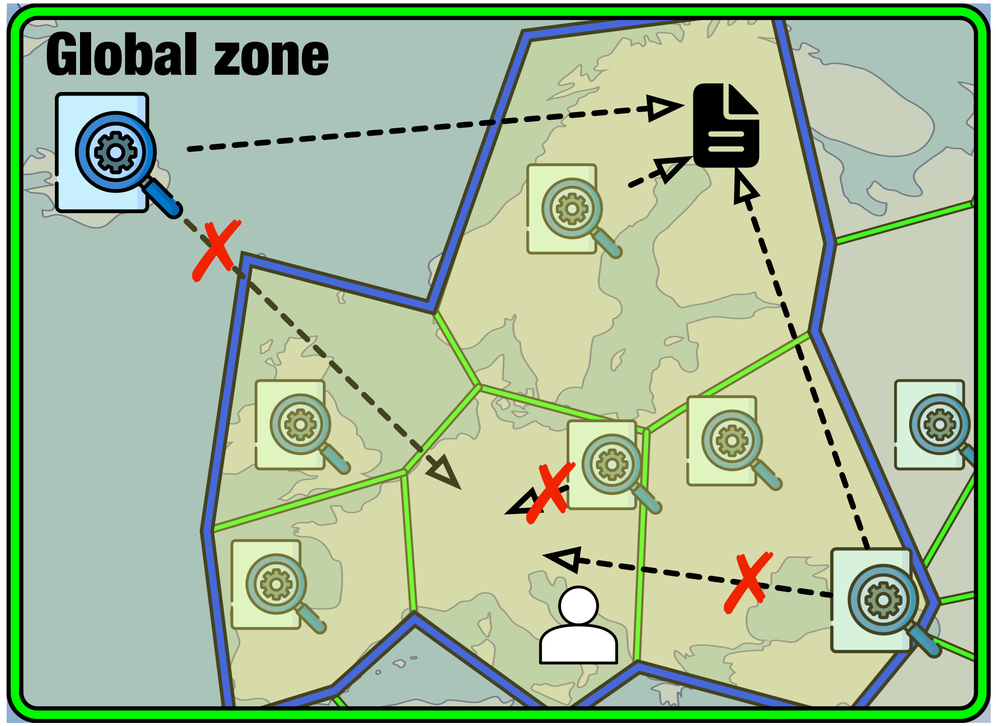
Locating latest data version during migration



Challenge 1: Inconsistent metadata during document migration

Solution: Update all relevant DS

Locating latest data version during migration

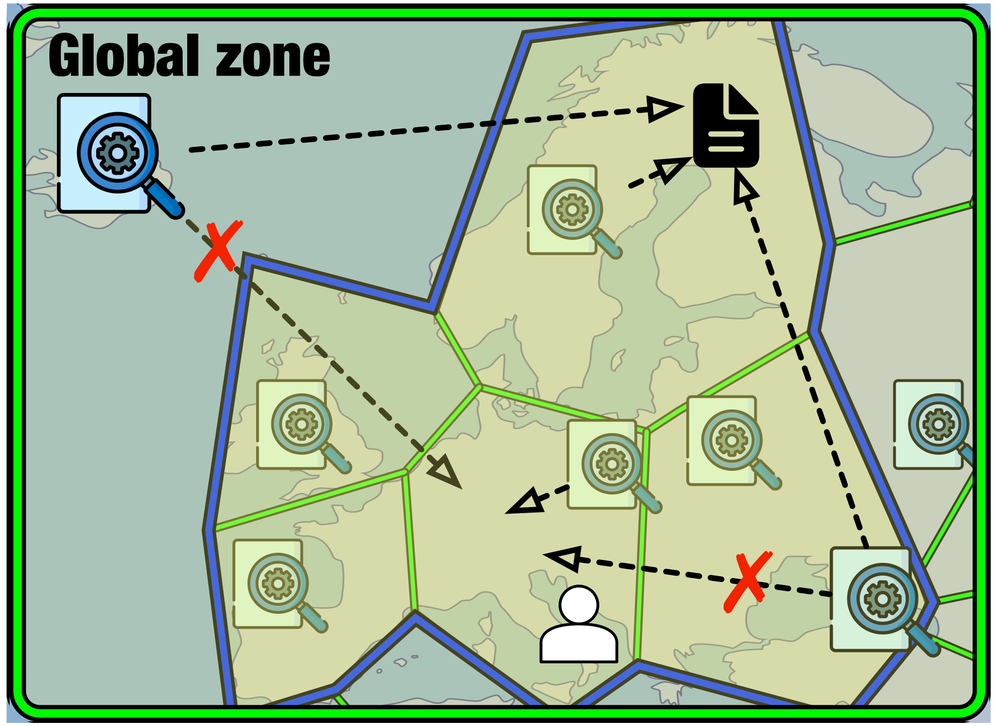


Challenge 1: Inconsistent metadata during document migration

Solution: Update all relevant DS

Challenge 2: Metadata updates not atomic

Locating latest data version during migration

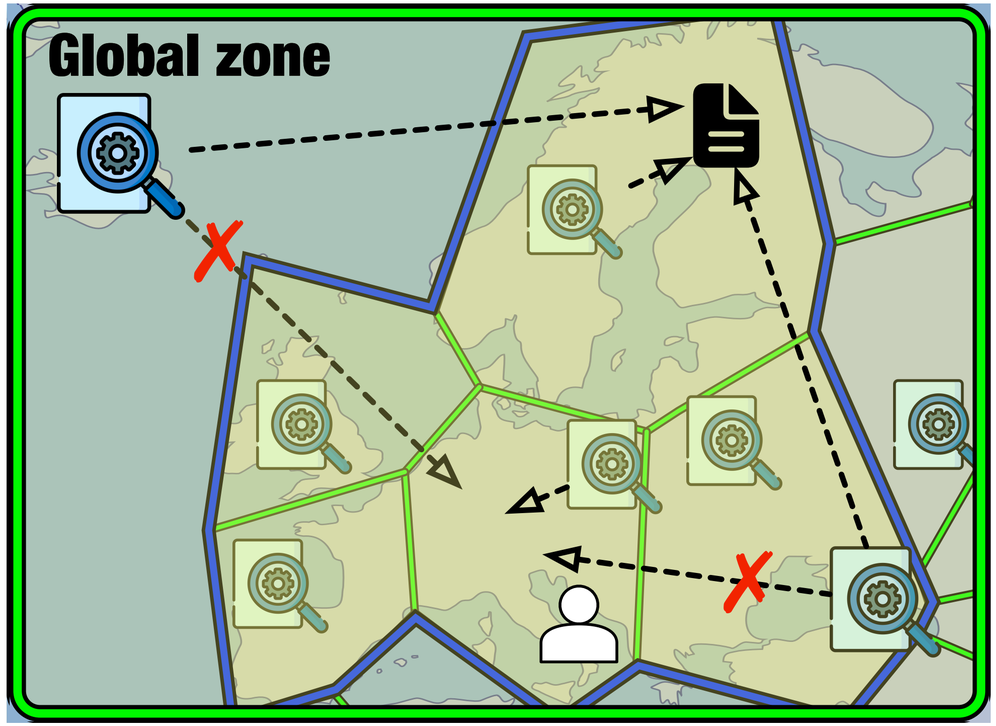


Challenge 1: Inconsistent metadata during document migration

Solution: Update all relevant DS

Challenge 2: Metadata updates not atomic

Locating latest data version during migration



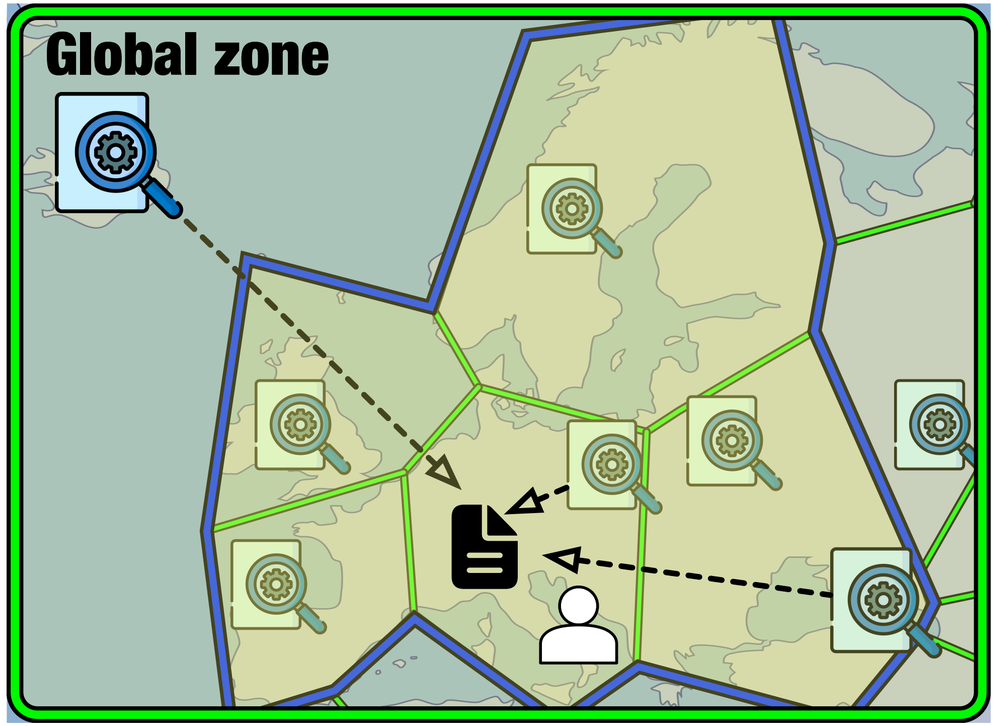
Challenge 1: Inconsistent metadata during document migration

Solution: Update all relevant DS

Challenge 2: Metadata updates not atomic

Solution: Store redirection marker before document migration

Locating latest data version during migration



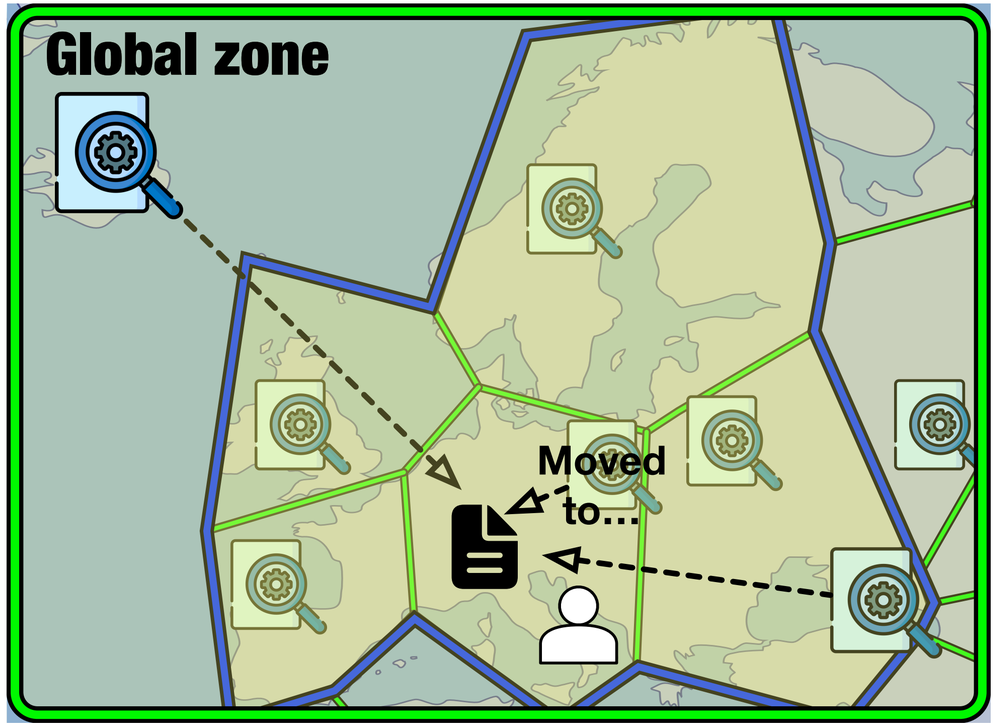
Challenge 1: Inconsistent metadata during document migration

Solution: Update all relevant DS

Challenge 2: Metadata updates not atomic

Solution: Store redirection marker before document migration

Locating latest data version during migration



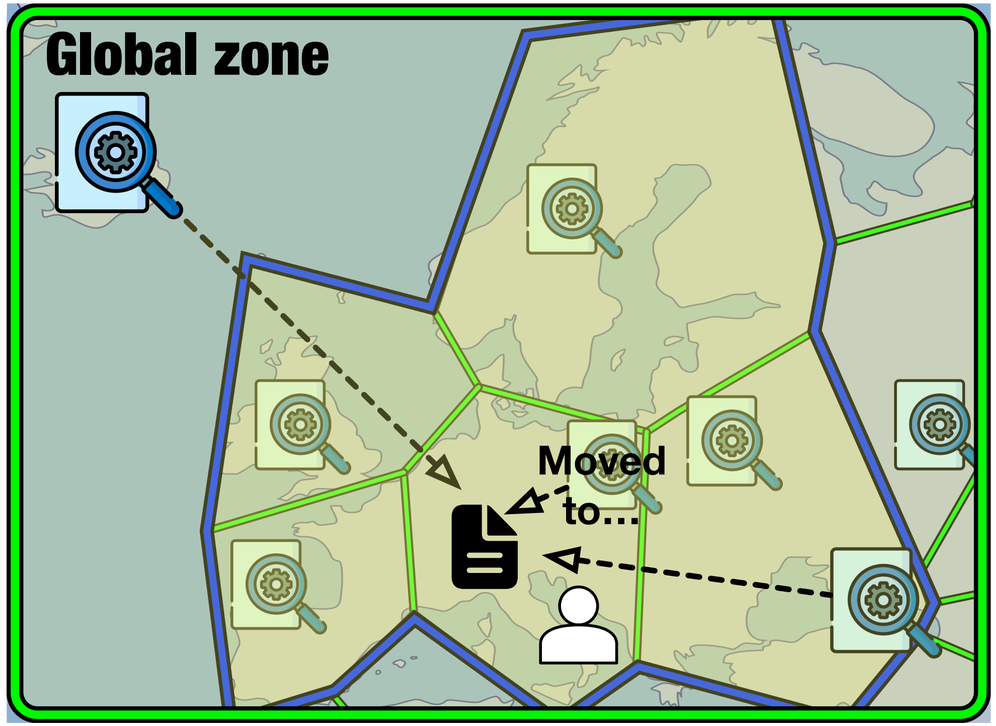
Challenge 1: Inconsistent metadata during document migration

Solution: Update all relevant DS

Challenge 2: Metadata updates not atomic

Solution: Store redirection marker before document migration

Locating latest data version during migration



Challenge 1: Inconsistent metadata during document migration

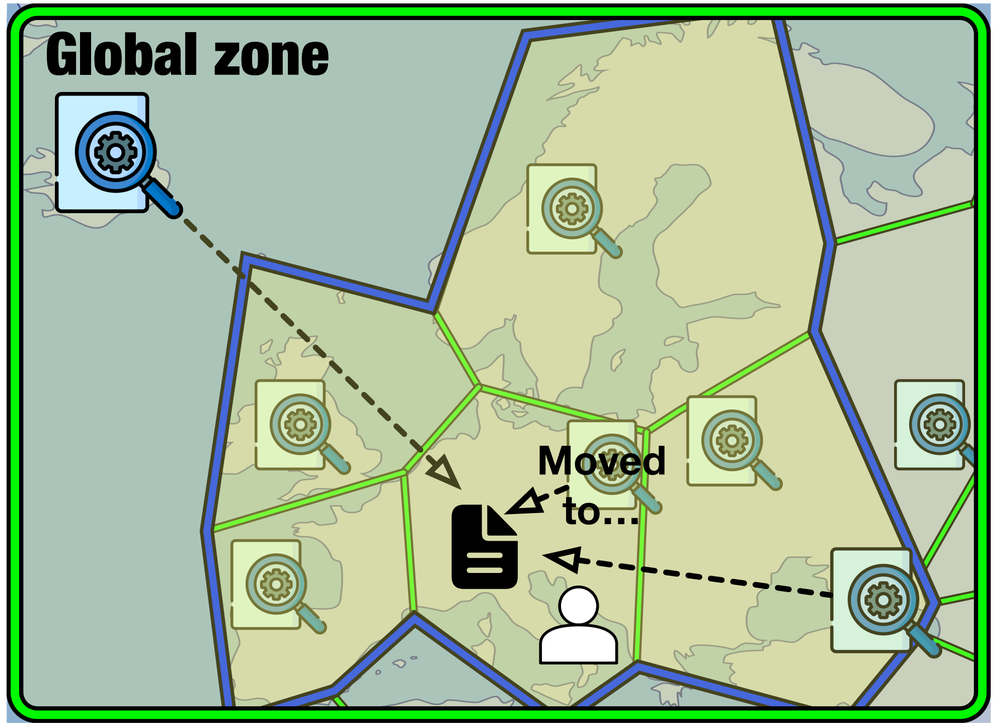
Solution: Update all relevant DS

Challenge 2: Metadata updates not atomic

Solution: Store redirection marker before document migration

Challenge 3: Redirection marker not atomic with migration

Locating latest data version during migration



Challenge 1: Inconsistent metadata during document migration

Solution: Update all relevant DS

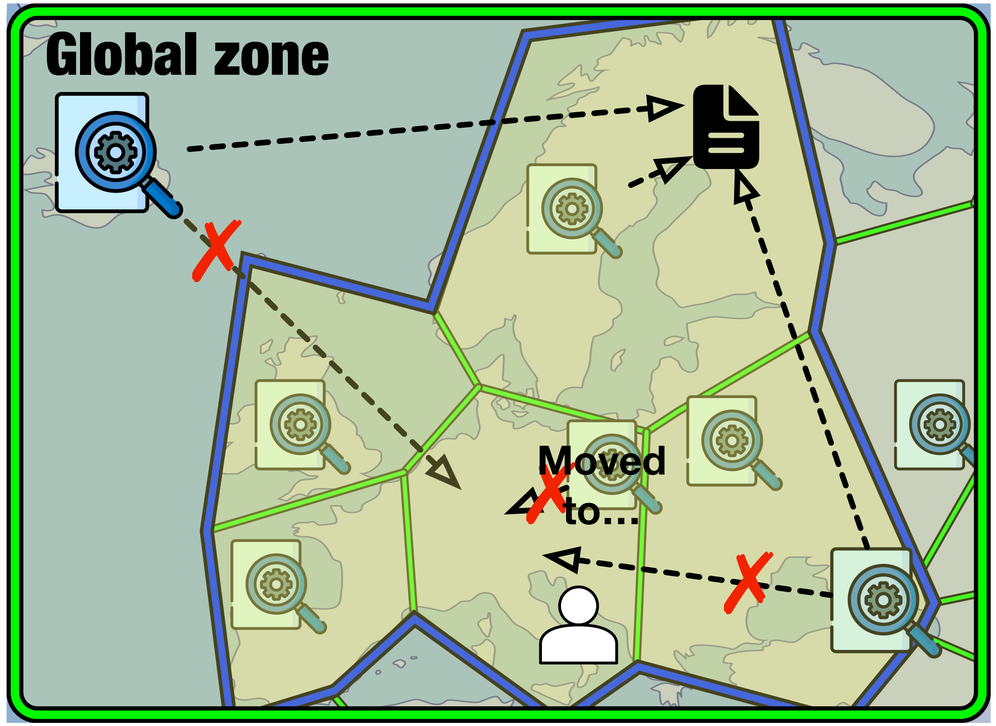
Challenge 2: Metadata updates not atomic

Solution: Store redirection marker before document migration

Challenge 3: Redirection marker not atomic with migration

Solution: Authoritative zone per item set via data plane upcalls

Locating latest data version during migration



Challenge 1: Inconsistent metadata during document migration

Solution: Update all relevant DS

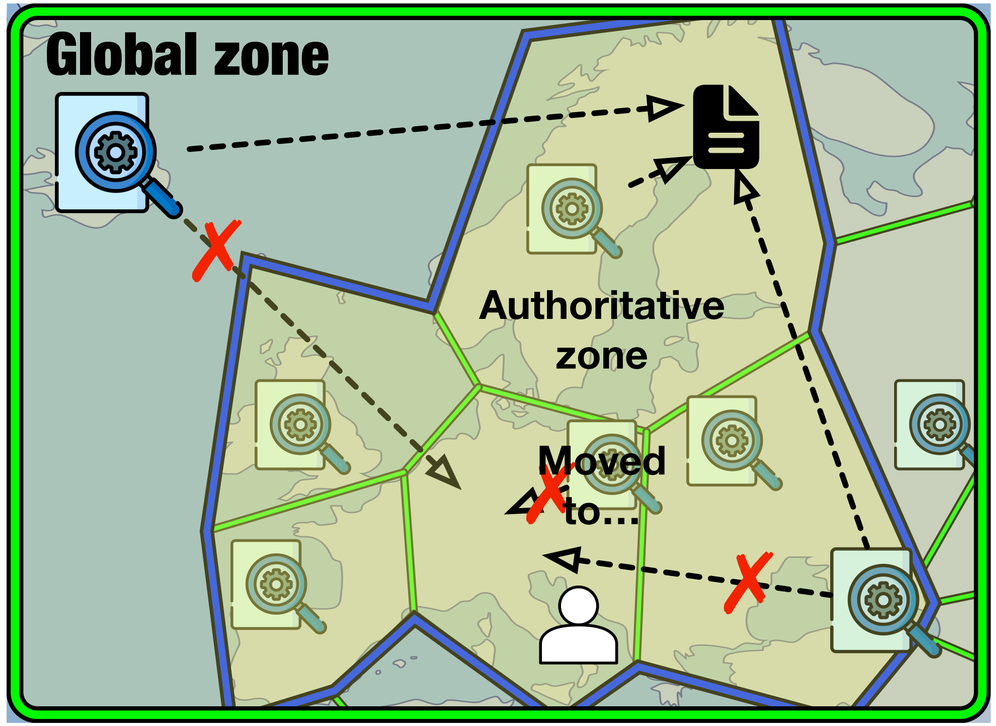
Challenge 2: Metadata updates not atomic

Solution: Store redirection marker before document migration

Challenge 3: Redirection marker not atomic with migration

Solution: Authoritative zone per item set via data plane upcalls

Locating latest data version during migration



Challenge 1: Inconsistent metadata during document migration

Solution: Update all relevant DS

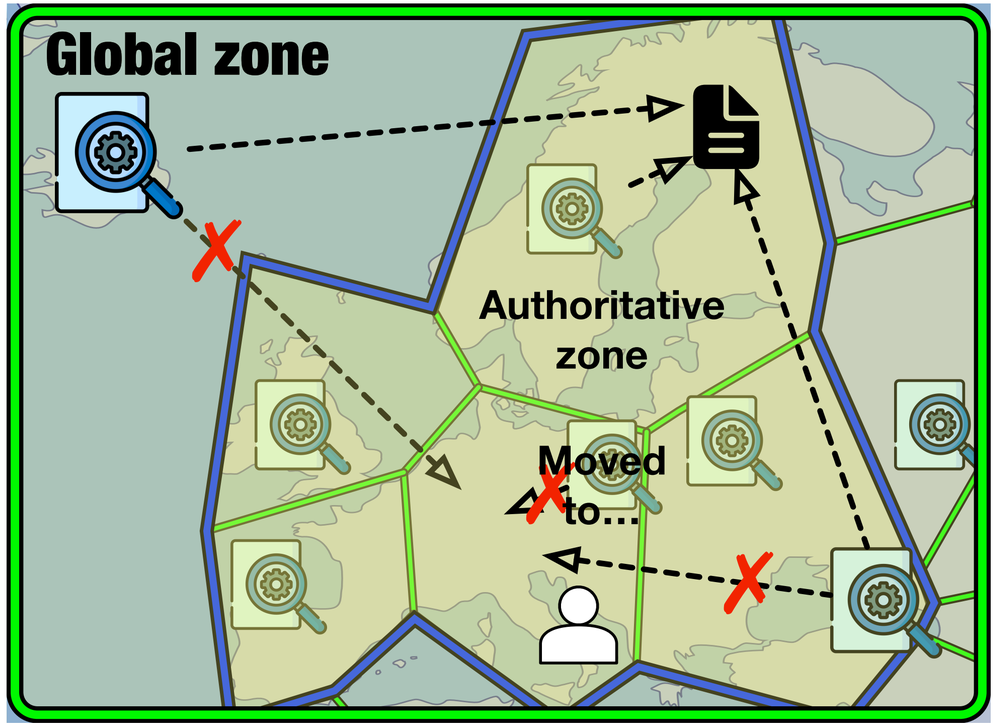
Challenge 2: Metadata updates not atomic

Solution: Store redirection marker before document migration

Challenge 3: Redirection marker not atomic with migration

Solution: Authoritative zone per item set via data plane upcalls

Locating latest data version during migration



Challenge 1: Inconsistent metadata during document migration

Solution: Update all relevant DS

Challenge 2: Metadata updates not atomic

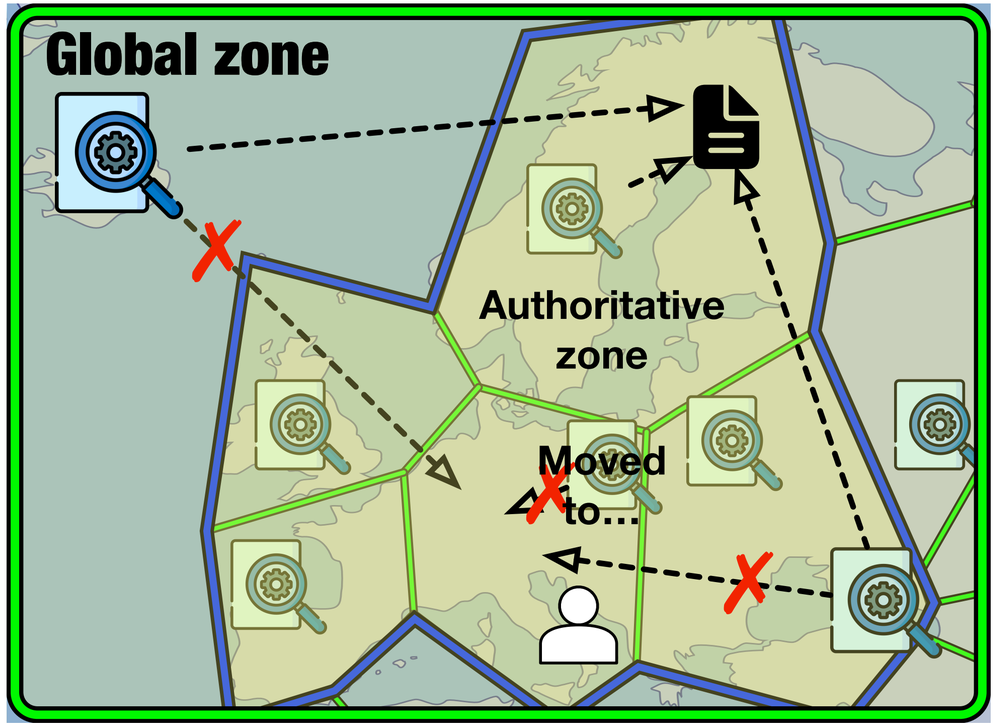
Solution: Store redirection marker before document migration

Challenge 3: Redirection marker not atomic with migration

Solution: Authoritative zone per item set via data plane upcalls

Challenge 4: Limit exposure for locating migrating docs

Locating latest data version during migration



Challenge 1: Inconsistent metadata during document migration

Solution: Update all relevant DS

Challenge 2: Metadata updates not atomic

Solution: Store redirection marker before document migration

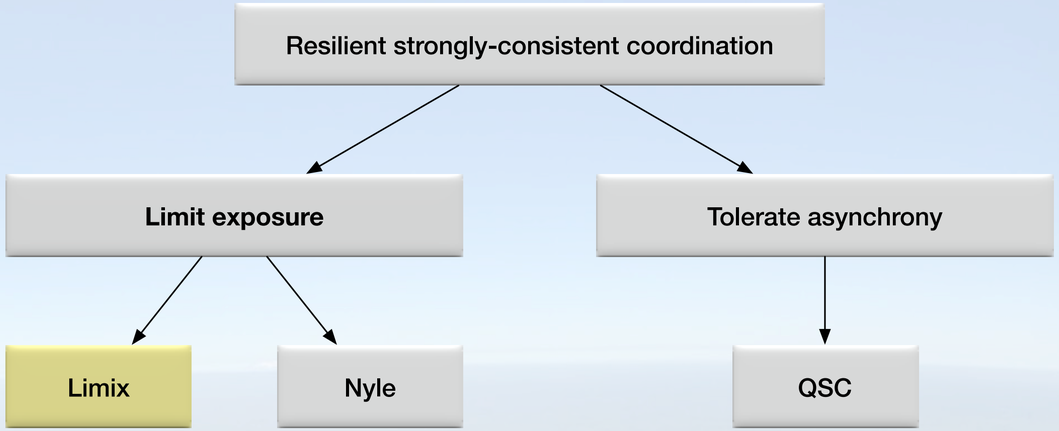
Challenge 3: Redirection marker not atomic with migration

Solution: Authoritative zone per item set via data plane upcalls

Challenge 4: Limit exposure for locating migrating docs

Solutions:
Pointer updates take place outside the critical path
Pointer versioning

Limix roadmap



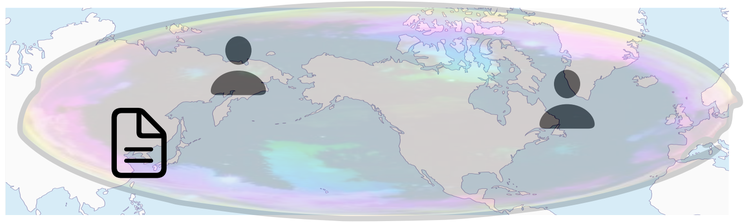
- The problem: metadata exposure
- Design of exposure-limiting metadata service
- Strongly-consistent item lookup
- Architecture
- **Evaluation**
  - **Jurisdictions: availability and costs per zone**
  - RTT-based autozoning: availability microbenchmark of Limix vs Physalia
  - RTT-based autozoning: realistic workloads Limix vs Physalia
- Summary

Administrative zoning: availability

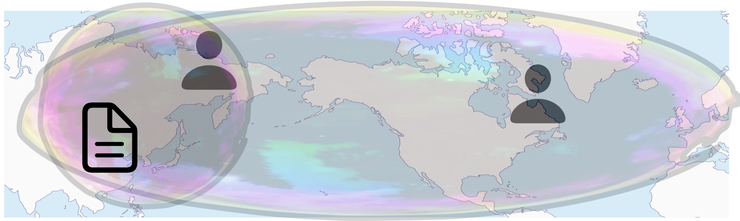
- Object administratively localized in a single region
- **Do local accesses succeed?**

Private cloud

Geo replication



Limix



Administrative zoning: availability

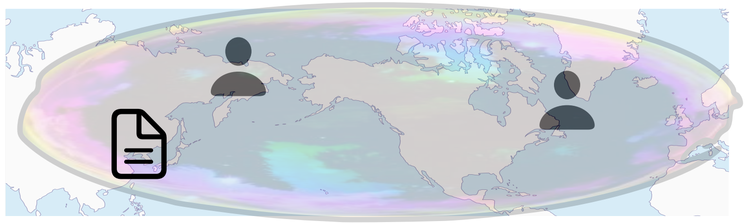
- Object administratively localized in a single region
- **Do local accesses succeed?**

Private cloud

~~Global manageability~~

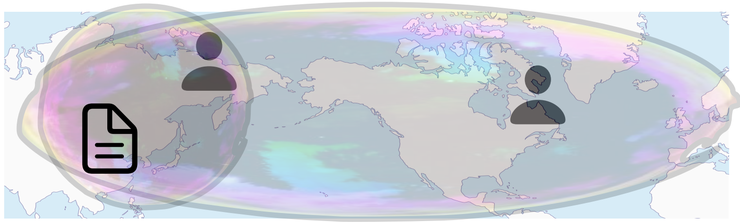
Geo replication

Global manageability



Limix

Global manageability

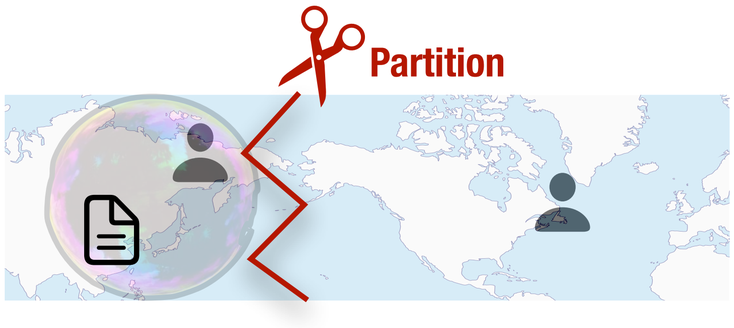


Administrative zoning: availability

- Object administratively localized in a single region
- **Do local accesses succeed?**

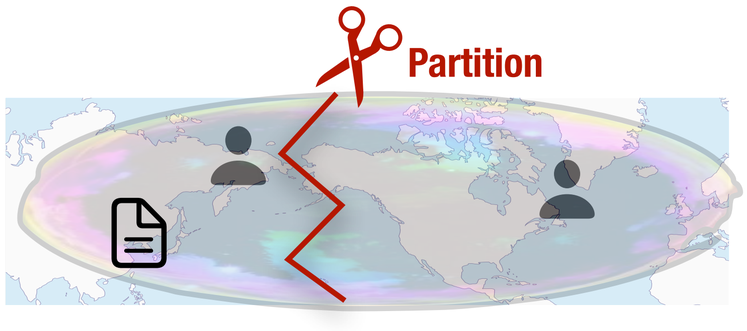
Private cloud

~~Global manageability~~



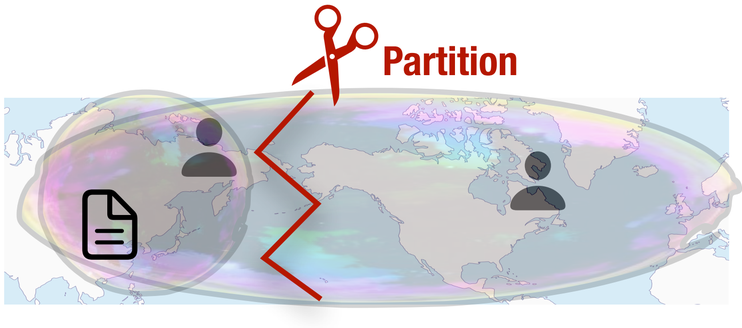
Geo replication

Global manageability



Limix

Global manageability

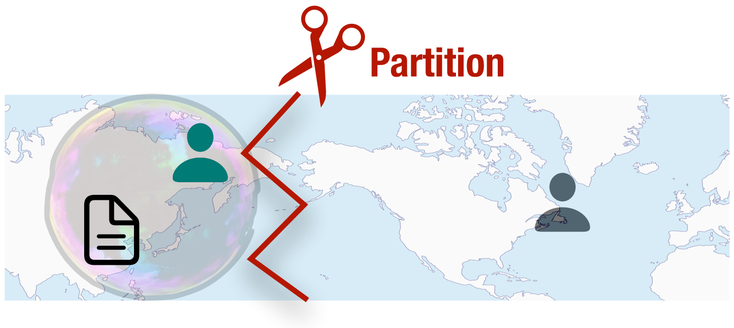


Administrative zoning: availability

- Object administratively localized in a single region
- **Do local accesses succeed?**

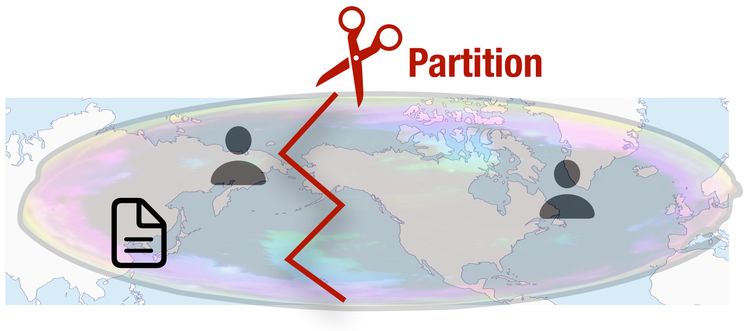
Private cloud

~~Global manageability~~



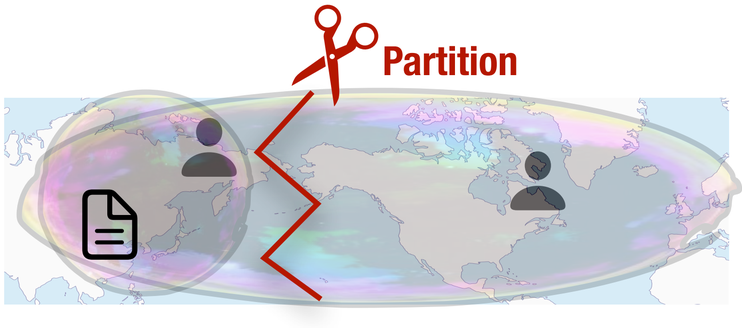
Geo replication

Global manageability



Limix

Global manageability



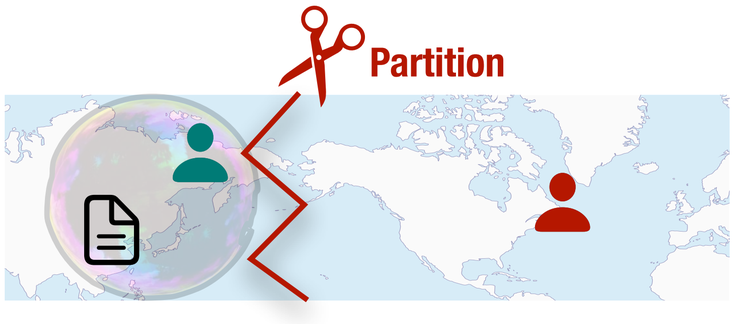
✔ Local access

Administrative zoning: availability

- Object administratively localized in a single region
- **Do local accesses succeed?**

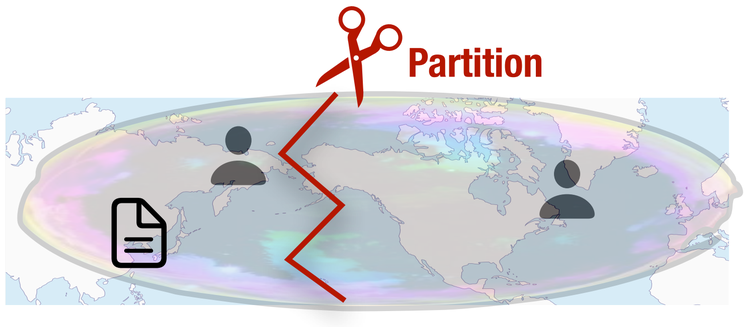
Private cloud

~~Global manageability~~



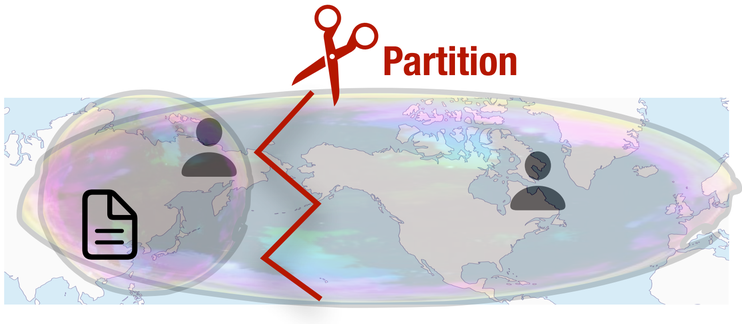
Geo replication

Global manageability



Limix

Global manageability



✔ Local access

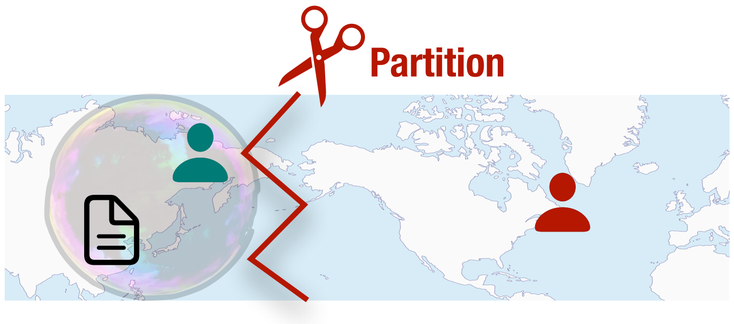
✗ Remote access

Administrative zoning: availability

- Object administratively localized in a single region
- **Do local accesses succeed?**

Private cloud

~~Global manageability~~

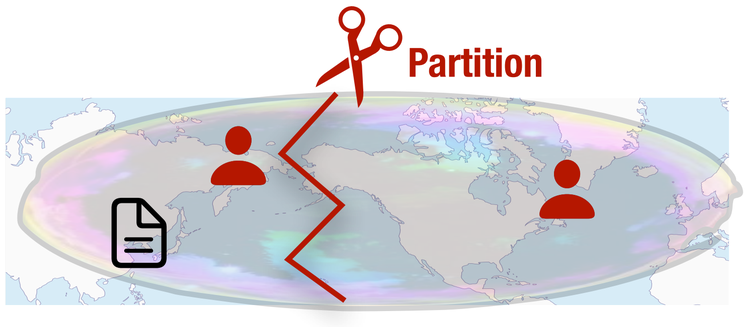


✔ Local access

✗ Remote access

Geo replication

Global manageability

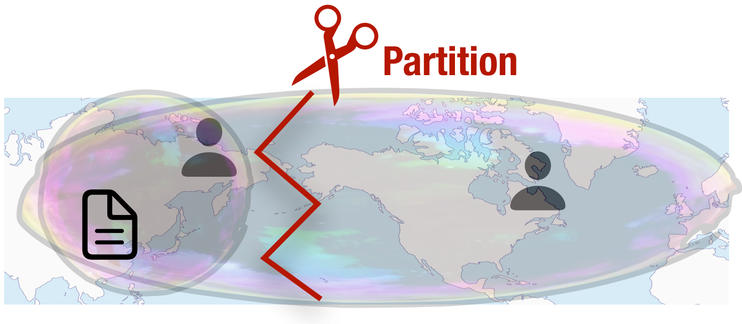


✗ Local access

✗ Remote access

Limix

Global manageability

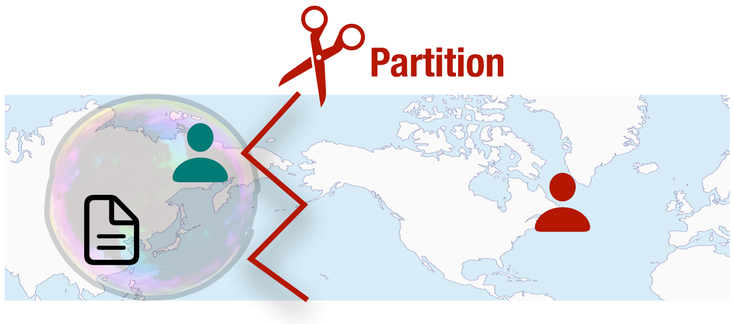


Administrative zoning: availability

- Object administratively localized in a single region
- **Do local accesses succeed?**

Private cloud

~~Global manageability~~

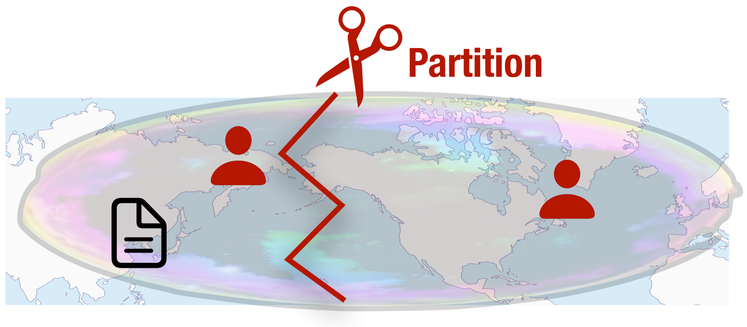


✔ Local access

✗ Remote access

Geo replication

Global manageability

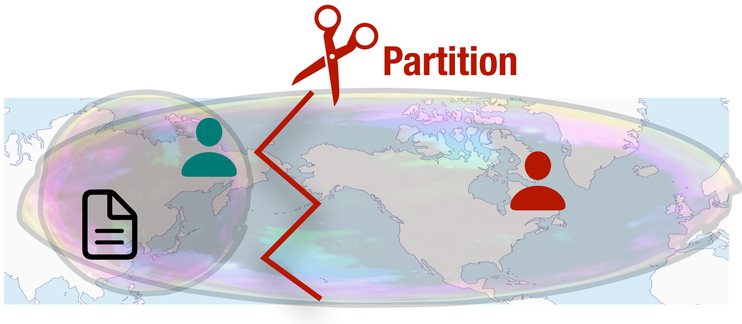


✗ Local access

✗ Remote access

Limix

Global manageability



✔ Local access

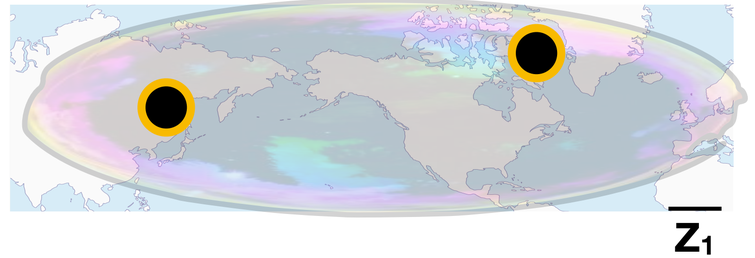
✗ Remote access

Administrative zoning: overhead

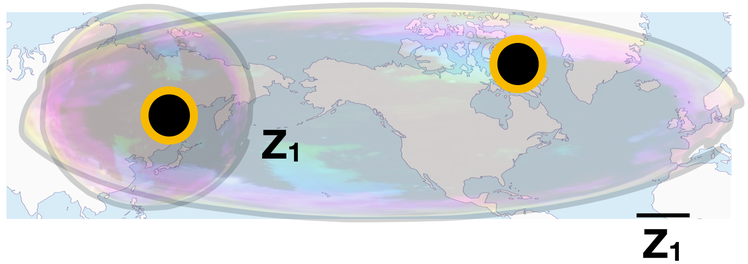
- **Cost per extra region deployed**
- Synthetic workload of 2000 ops (Limix: 1000 x 2 zones)

Private cloud

Geo replication

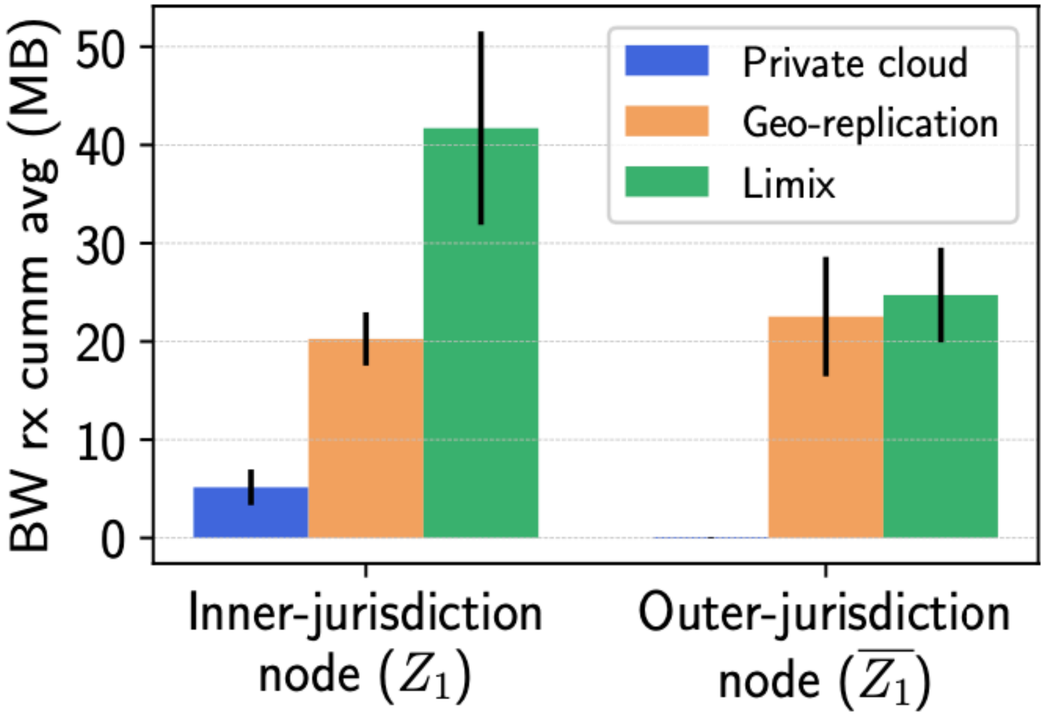


Limix



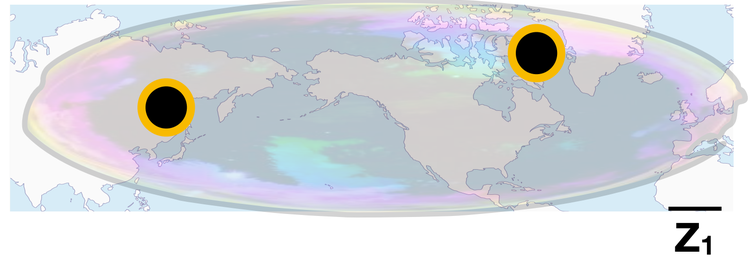
Administrative zoning: overhead

- **Cost per extra region deployed**
- Synthetic workload of 2000 ops (Limix: 1000 x 2 zones)

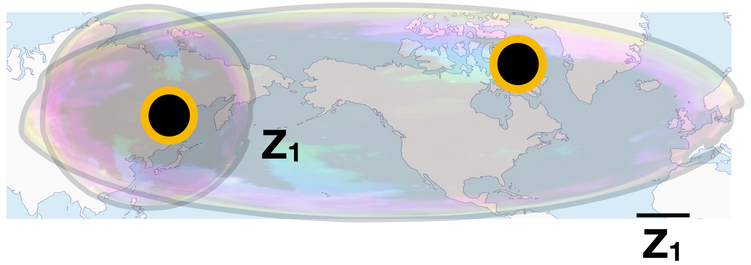


Private cloud

Geo replication

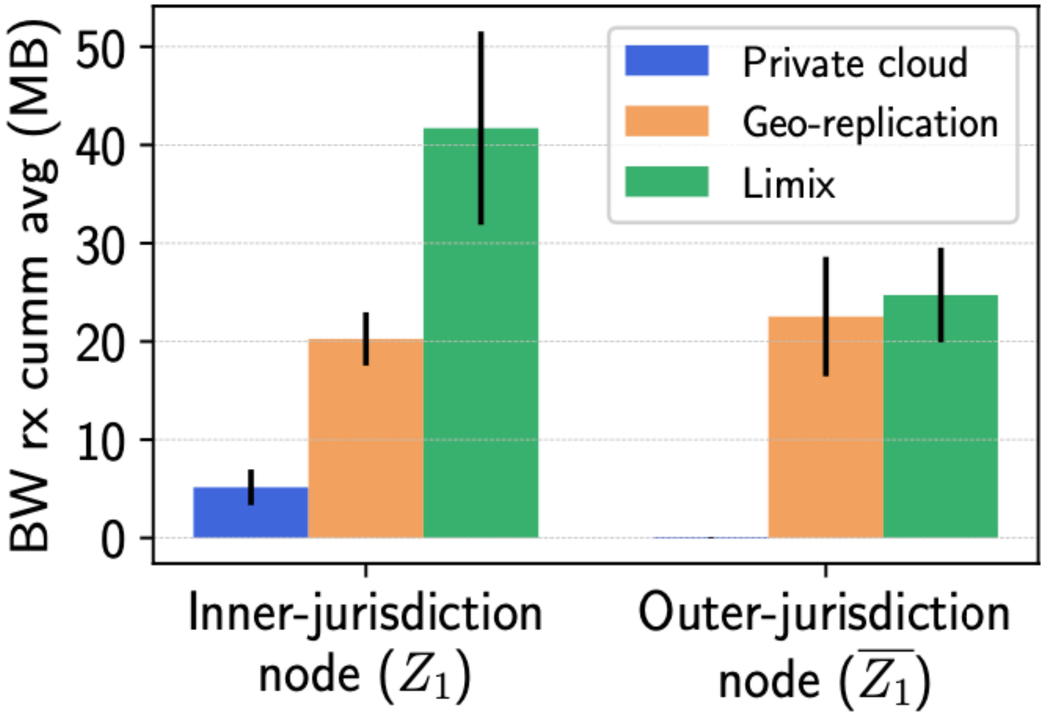


Limix



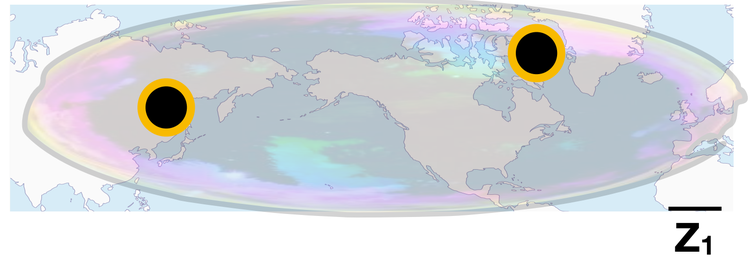
Administrative zoning: overhead

- **Cost per extra region deployed**
- Synthetic workload of 2000 ops (Limix: 1000 x 2 zones)

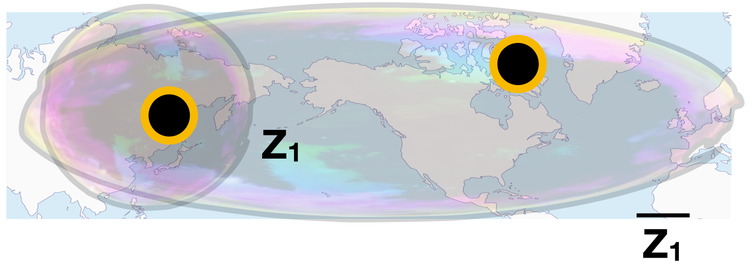


Private cloud

Geo replication

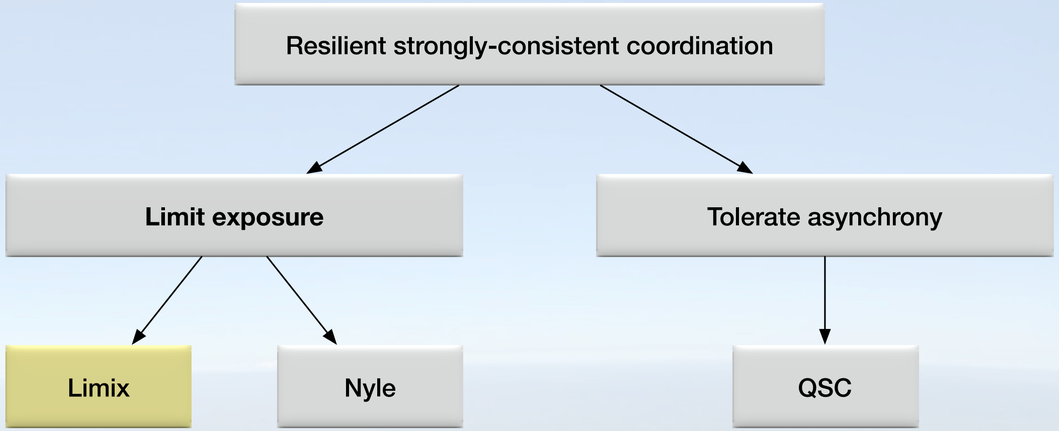


Limix



Suitable for a “pay-as-you-go” model

Limix roadmap



- The problem: exposure of data lookup
- Design of exposure-limiting metadata service
- Strongly-consistent item lookup
- Architecture
- **Evaluation**
  - Jurisdictions: availability and costs per zone
  - **RTT-based autozoning: microbenchmark of Limix vs Physalia**
  - RTT-based autozoning: realistic workloads Limix vs Physalia
- Summary

Limix vs Physalia setup

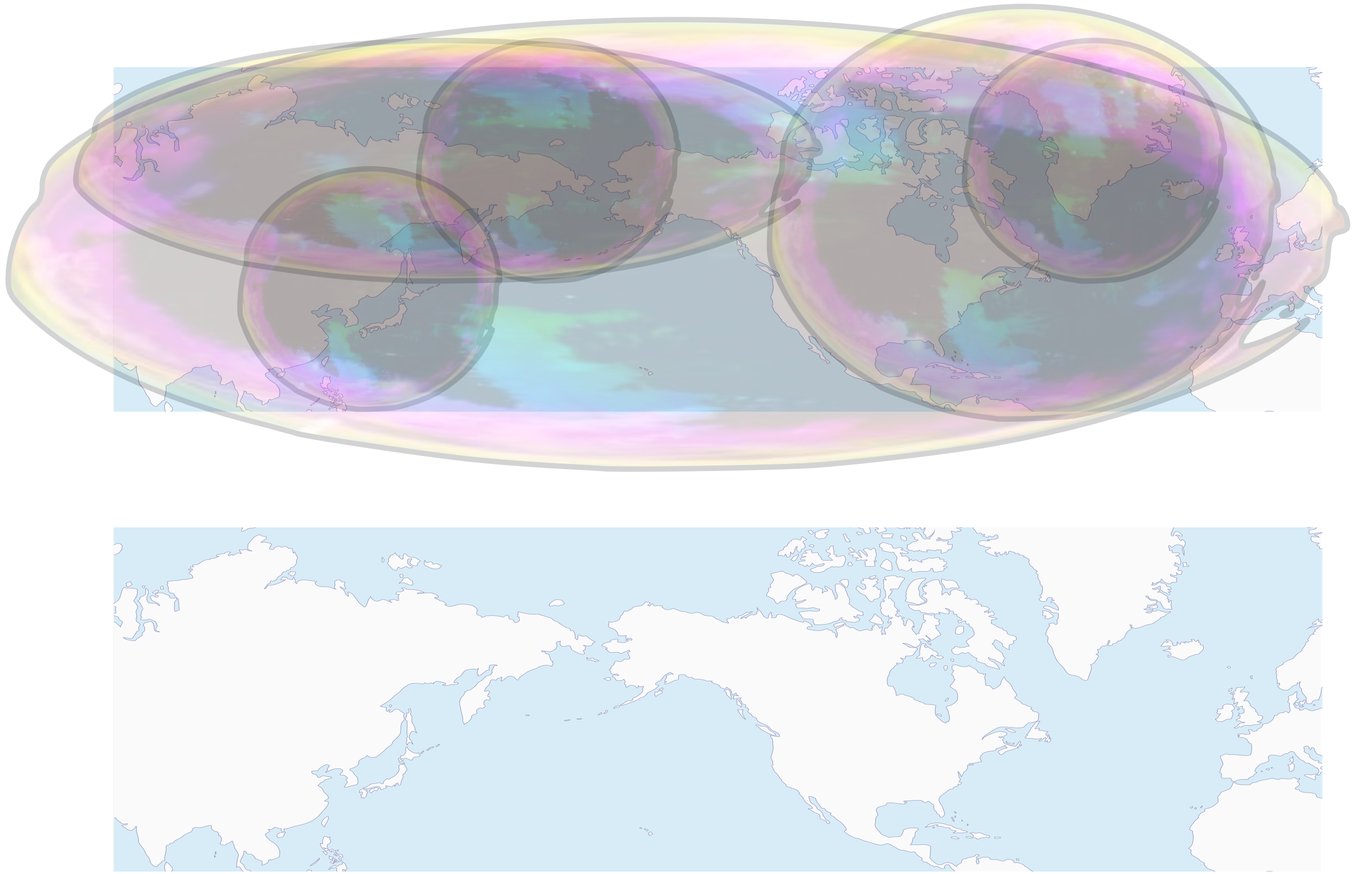
- **Do far away failures affect accesses?**

- **20 AWS sites**: N America, Europe and Asia-Pacific; RTTs [0.43, 250.31] ms

Limix vs Physalia setup

- **Do far away failures affect accesses?**

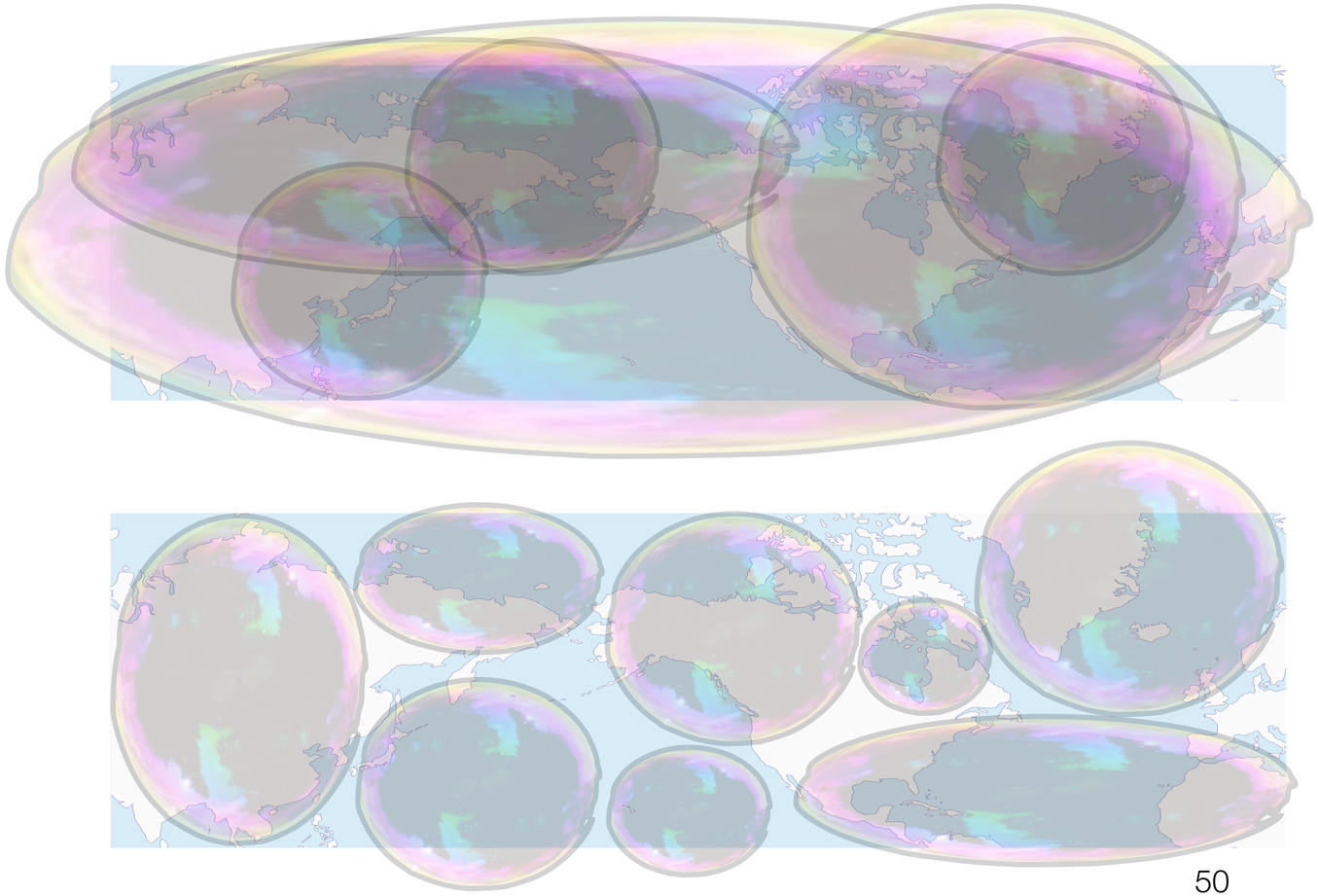
- **20 AWS sites**: N America, Europe and Asia-Pacific; RTTs [0.43, 250.31] ms



- **Limix RTT-based autozoning**

Limix vs Physalia setup

- **Do far away failures affect accesses?**
- **20 AWS sites**: N America, Europe and Asia-Pacific; RTTs [0.43, 250.31] ms



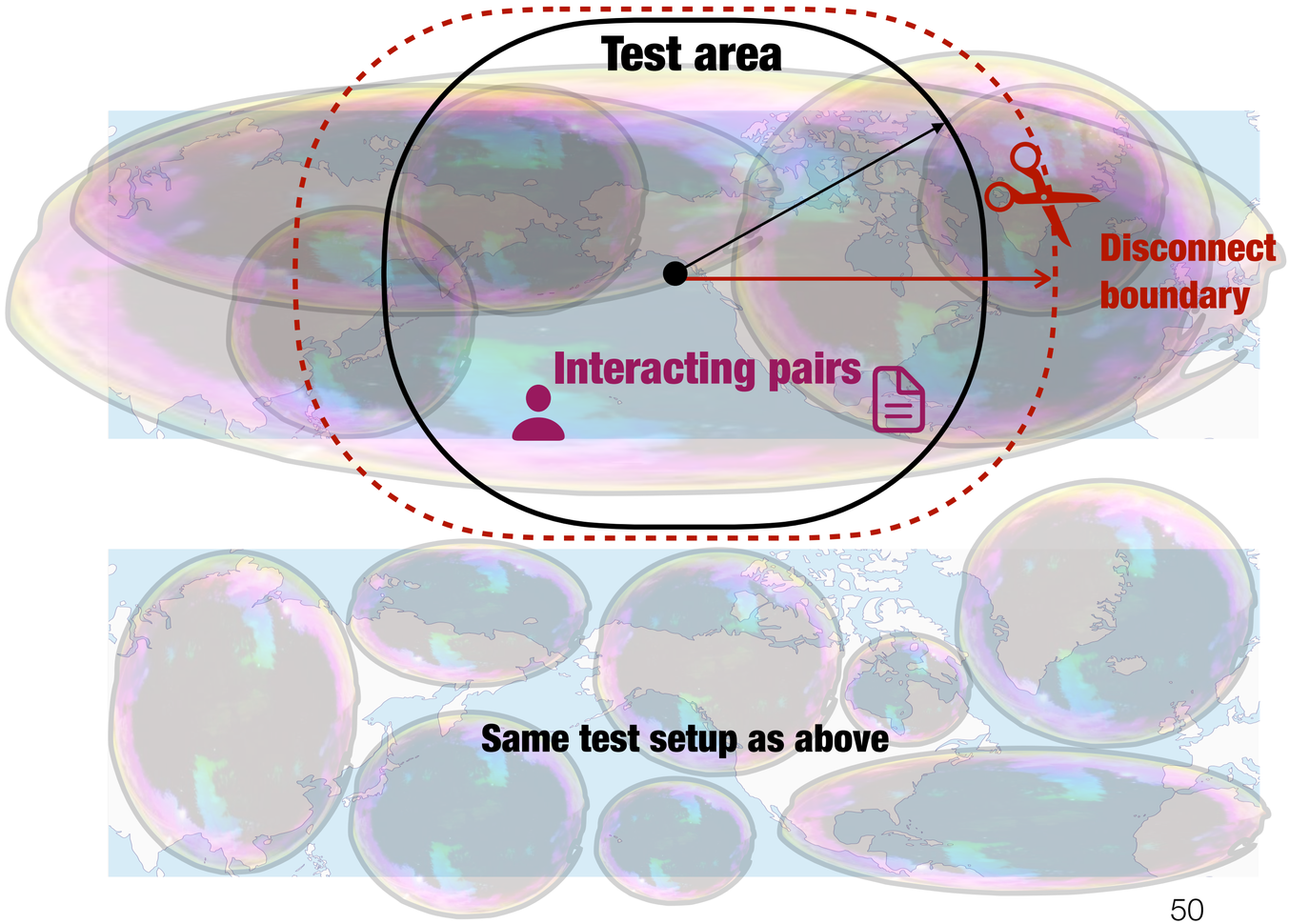
- **Limix RTT-based autozoning**
- **Physalia***
  - Cells of max 50ms RTT diameter

*Marc Brooker, Tao Chen, and Fan Ping. Millions of tiny databases. In *Conference on Networked Systems Design and Implementation (NSDI)*, 2020.

Limix vs Physalia setup

- **Do far away failures affect accesses?**

- **20 AWS sites**: N America, Europe and Asia-Pacific; RTTs [0.43, 250.31] ms

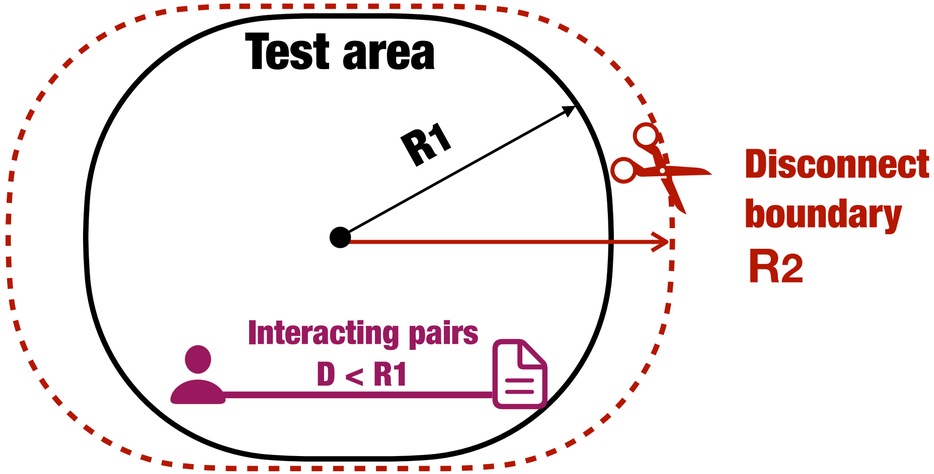


- **Limix RTT-based autozoning**

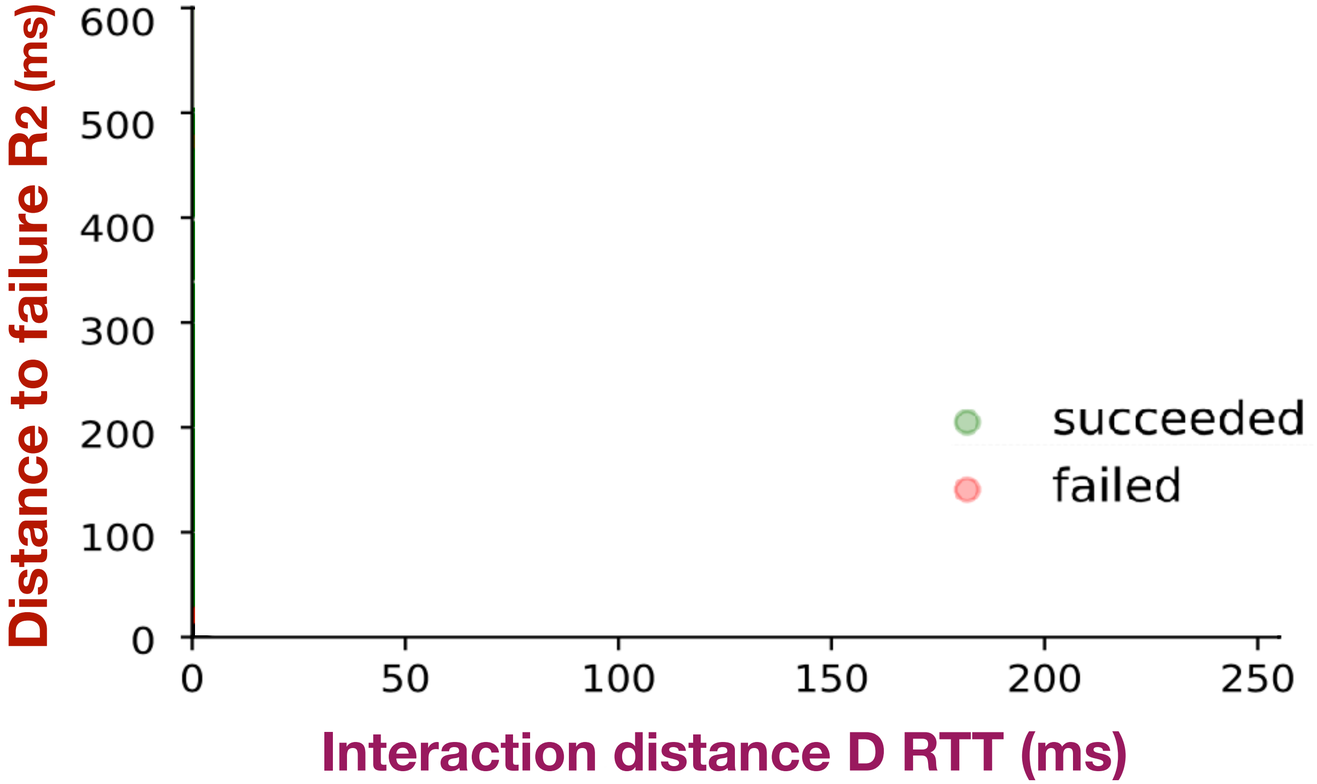
- **Physalia***
  - Cells of max 50ms RTT diameter

*Marc Brooker, Tao Chen, and Fan Ping. Millions of tiny databases. In *Conference on Networked Systems Design and Implementation (NSDI)*, 2020.

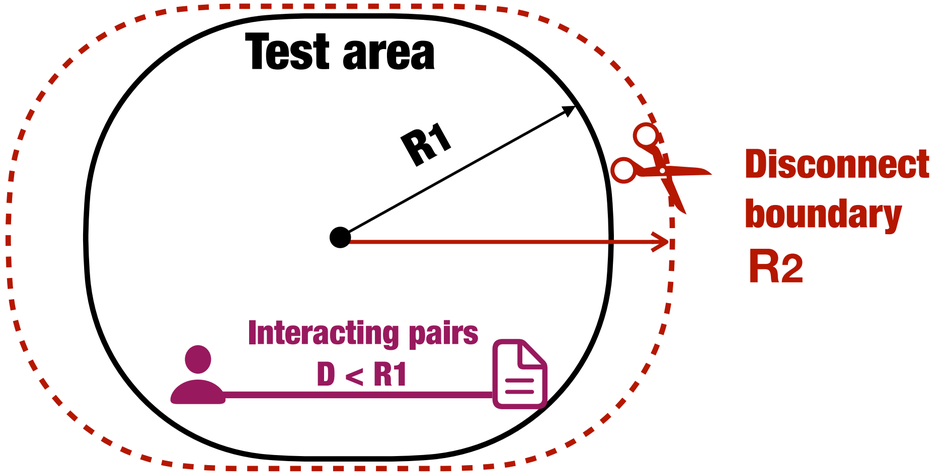
Limix vs Physalia: availability



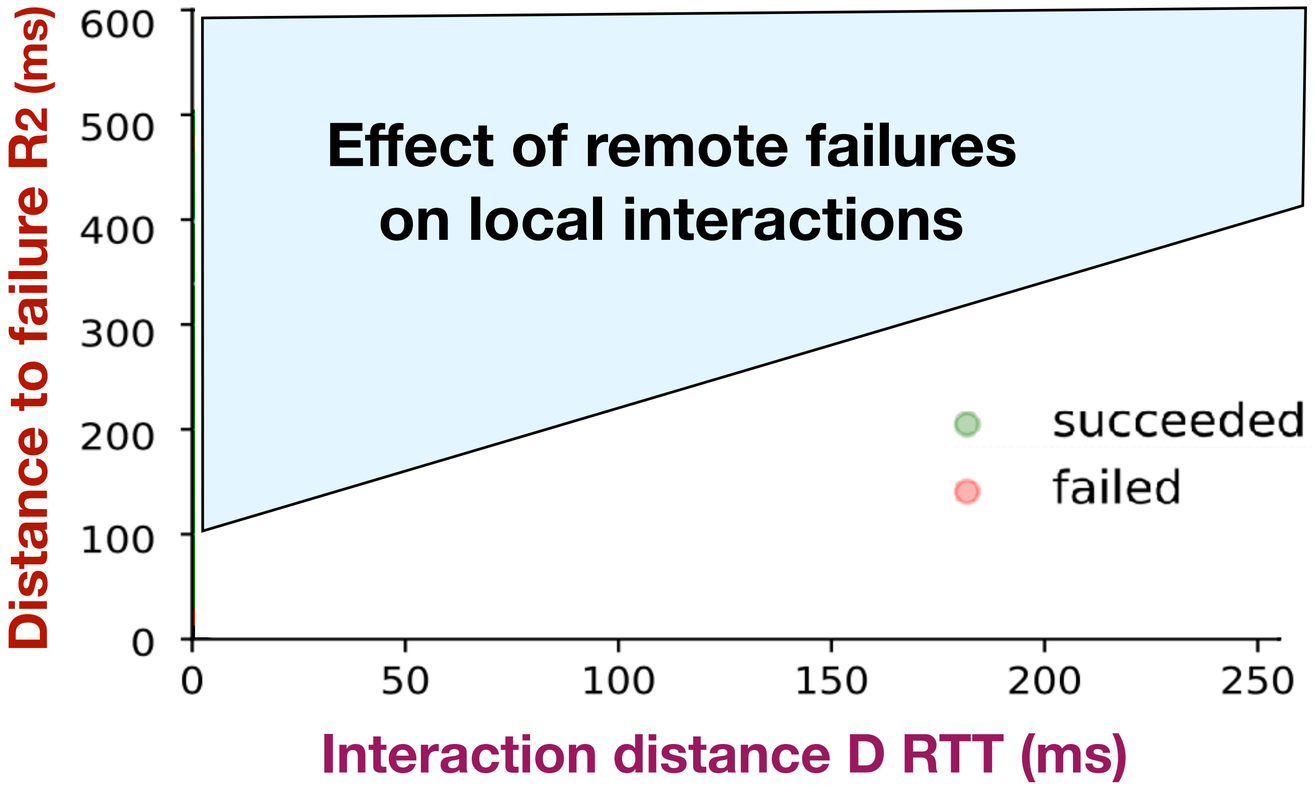
- **Workload: per R1** 1000 rnd pairs



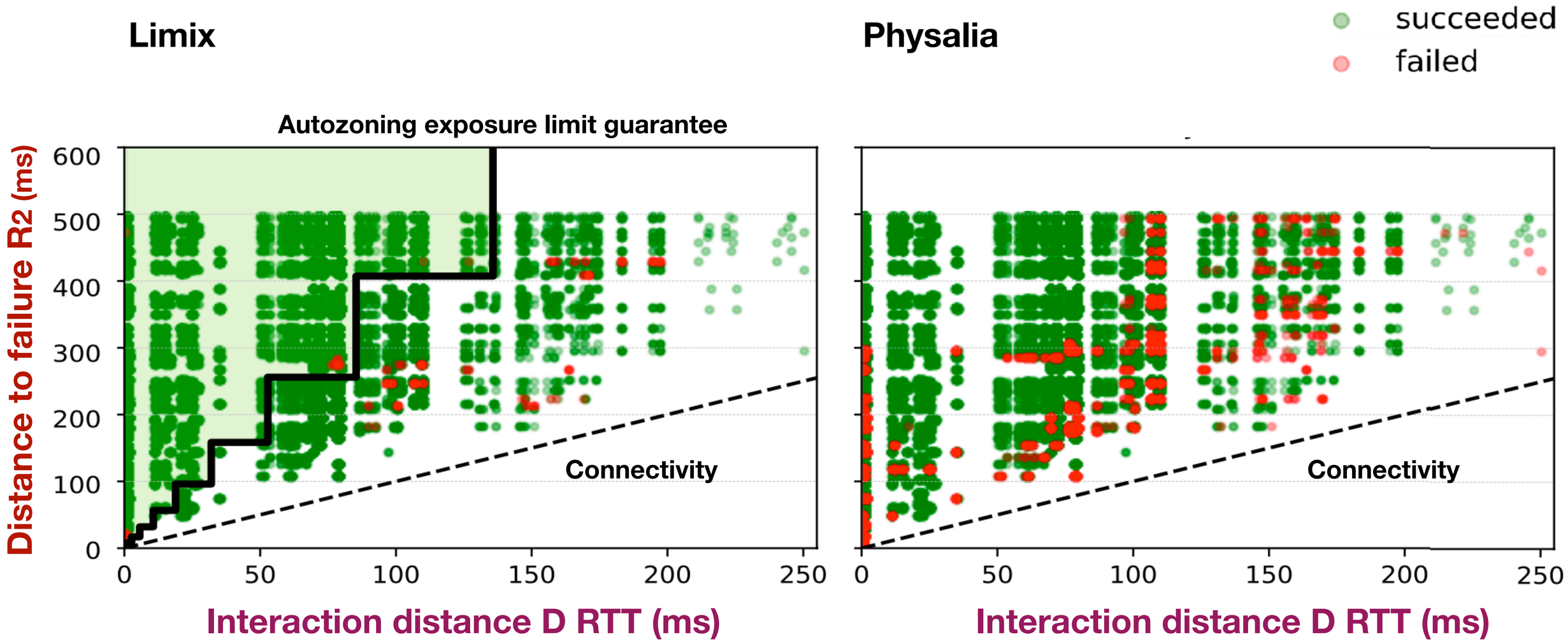
Limix vs Physalia: availability



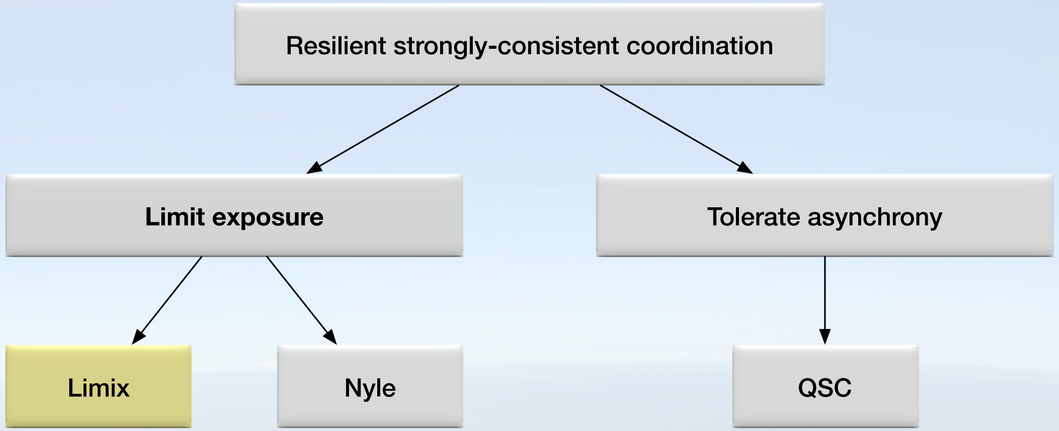
- **Workload: per R1** 1000 rnd pairs



Limix vs Physalia: availability



Limix roadmap

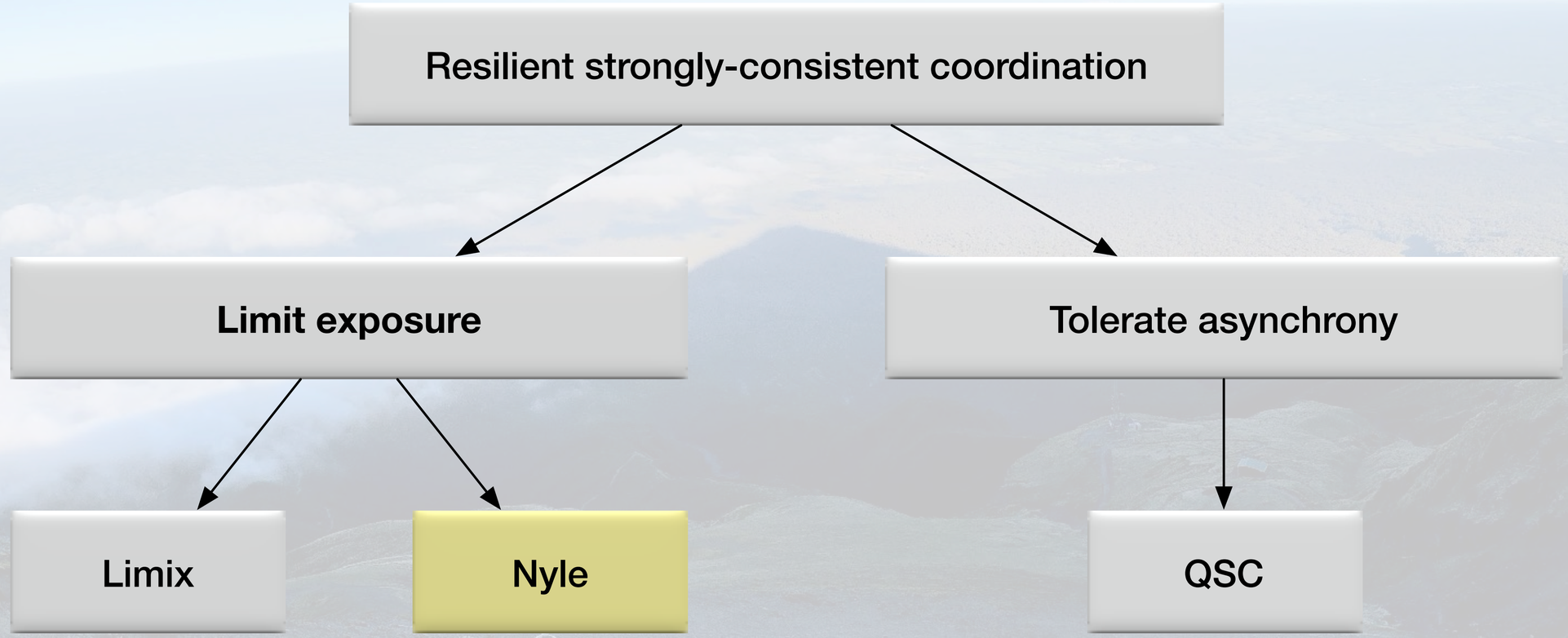


- The problem: exposure of data lookup
- Design of exposure-limiting metadata service
- Strongly-consistent item lookup
- Architecture
- Evaluation
- **Summary**

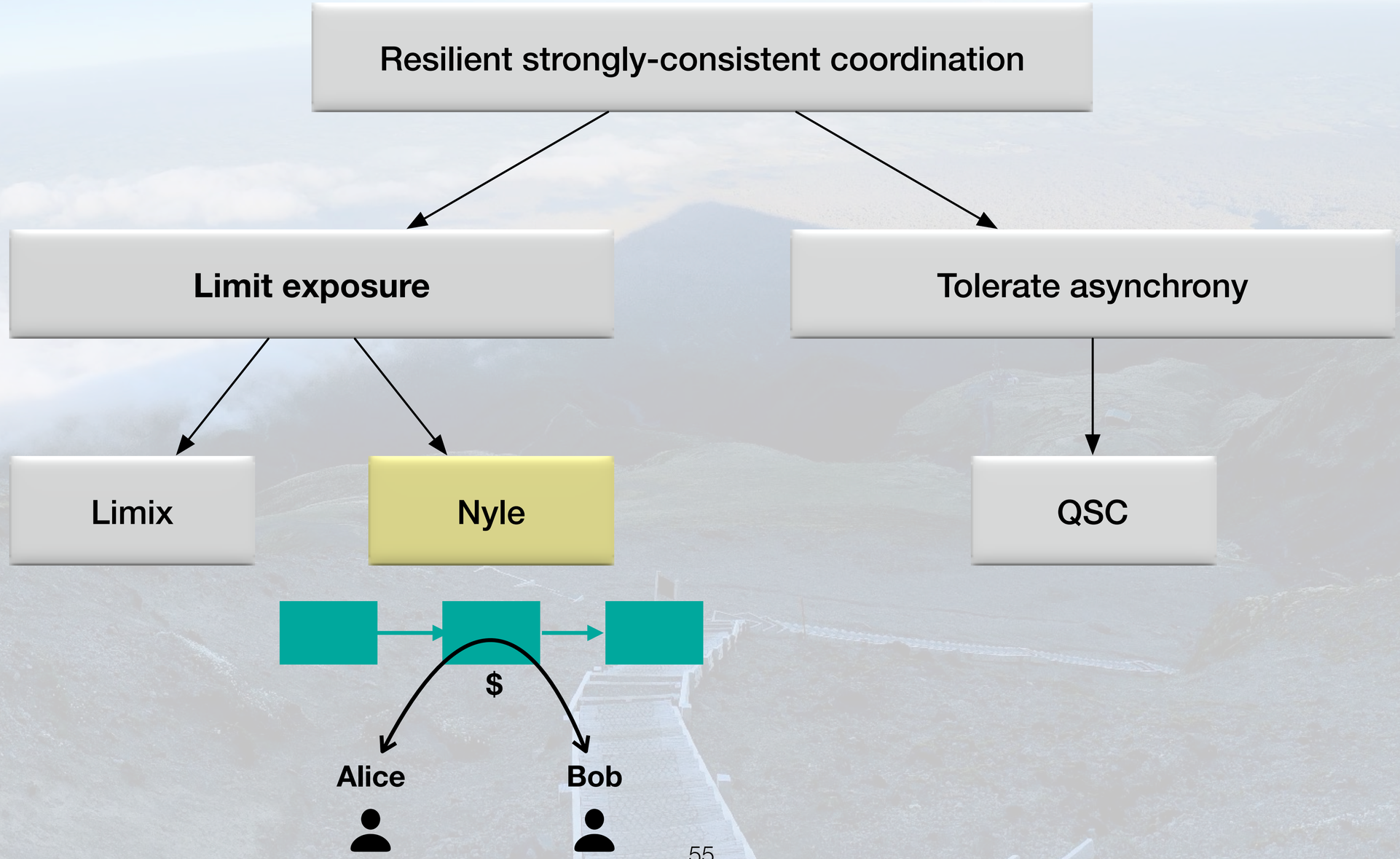
Limix summary

- **Global metadata service that protects local accesses from distant failures**
  - Suitable for strongly consistent data planes
- **Pay-as-you-go administrative zone deployment**
- **RTT-based autozoning outperforms related work**

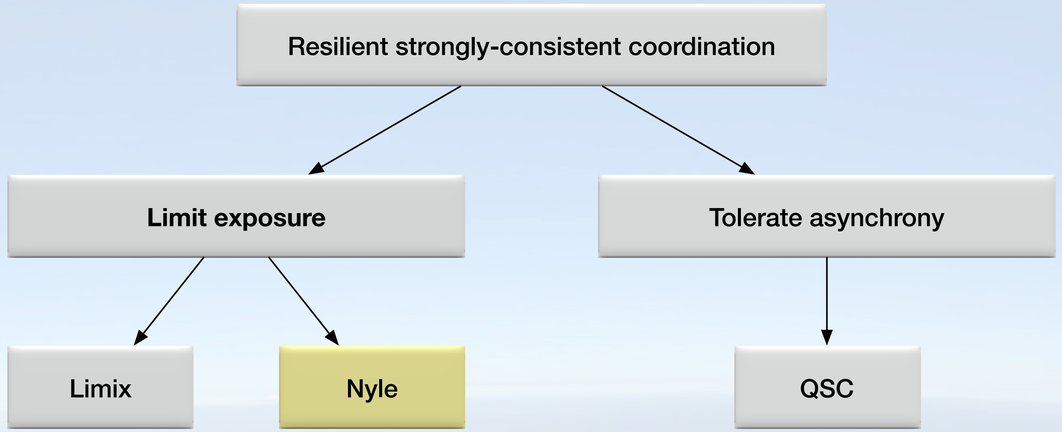
Thesis roadmap



Thesis roadmap



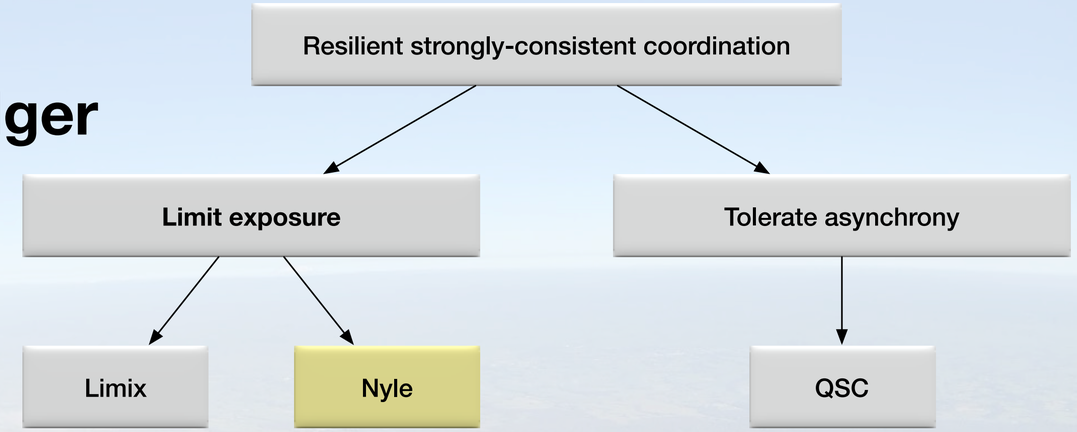
Nyle roadmap



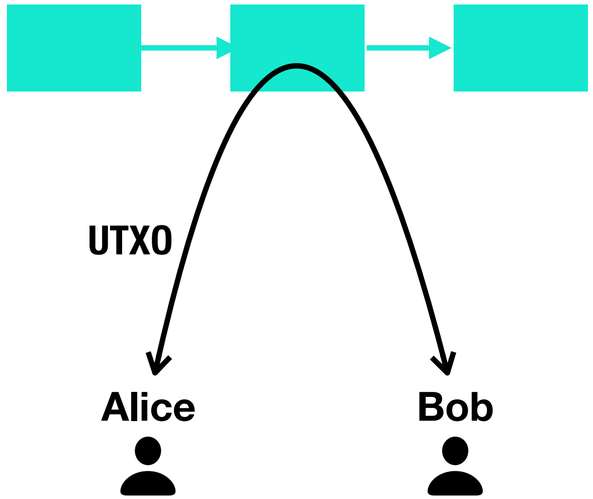
- The problem: exposure-limiting distributed ledger
- Secure zoning with Byzantine validators
- A trust-but-verify design for transaction validation
- Multi-input multi-output transactions
- Clearance zone change and censorship attacks
- Simulation
- Summary

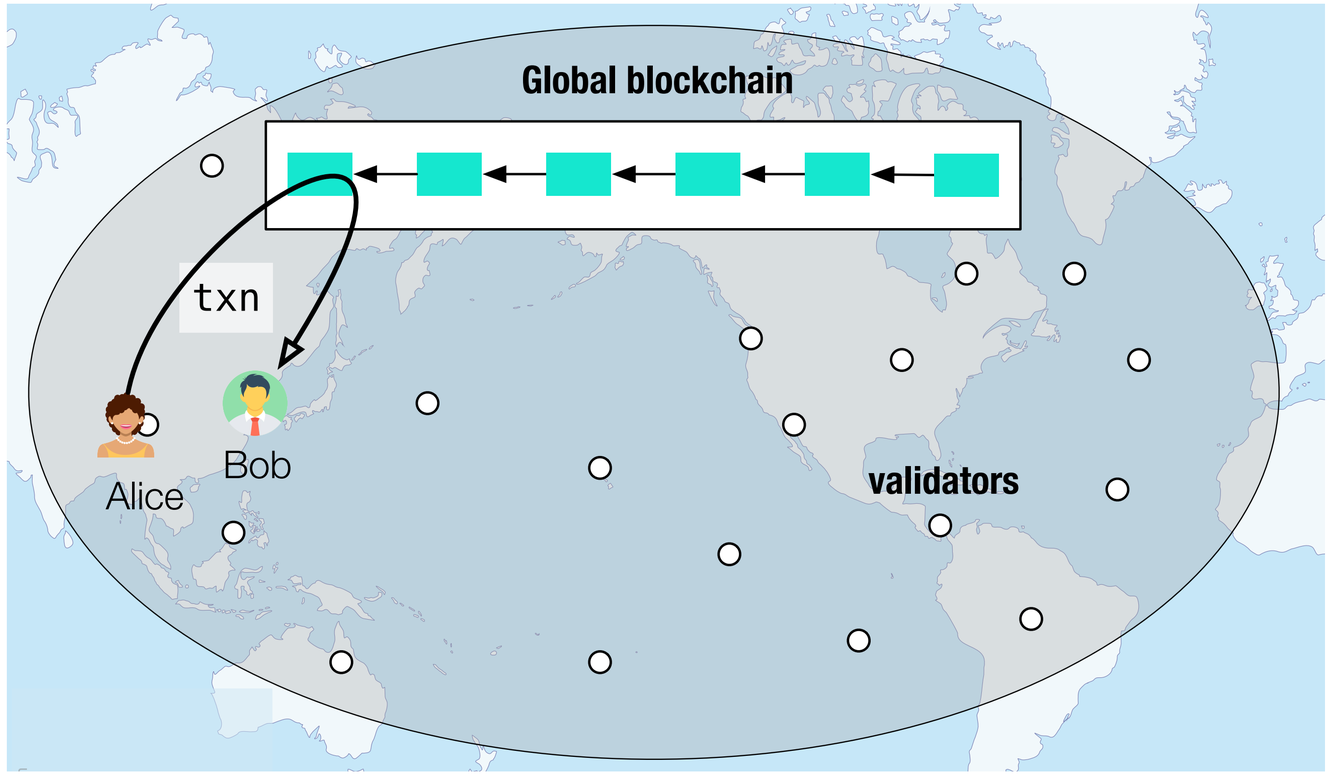
Nyle roadmap

- **The problem: exposure-limiting distributed ledger**
  - Motivation
  - Setup and goals
- Secure zoning with Byzantine validators
- A trust-but-verify design for transaction validation
- Multi-input multi-output transactions
- Clearance zone change and censorship attacks
- Simulation
- Summary

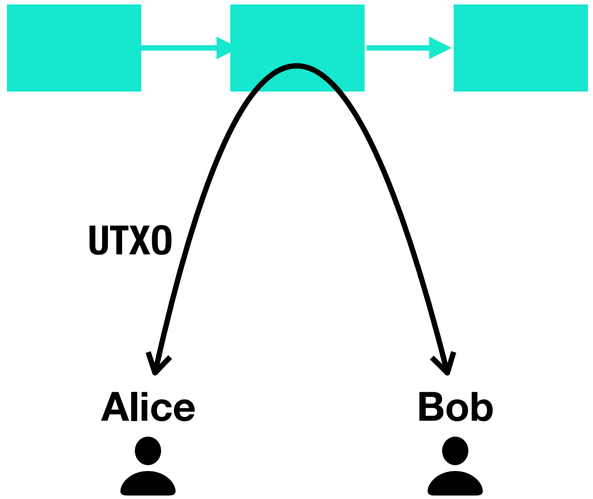


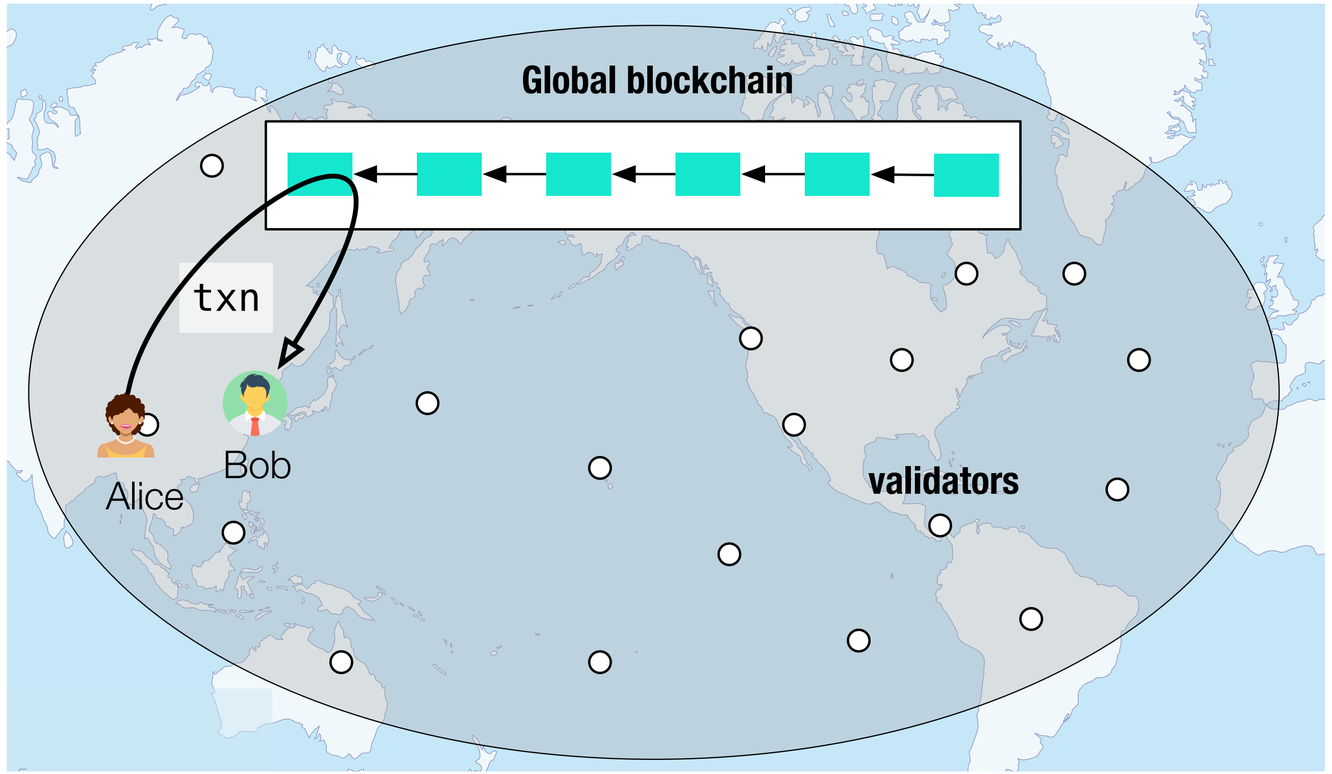
Distributed ledger





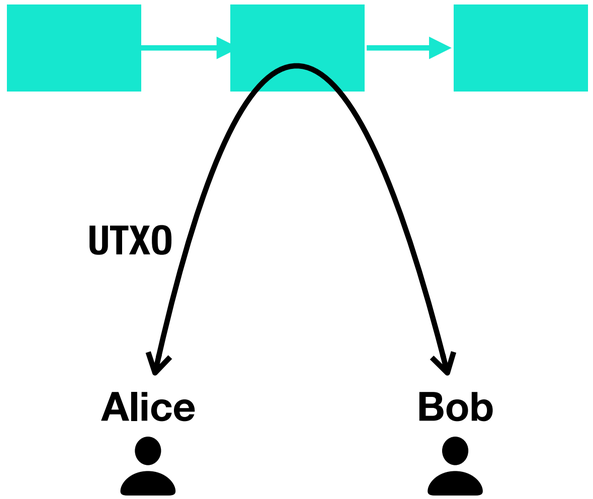
Distributed ledger

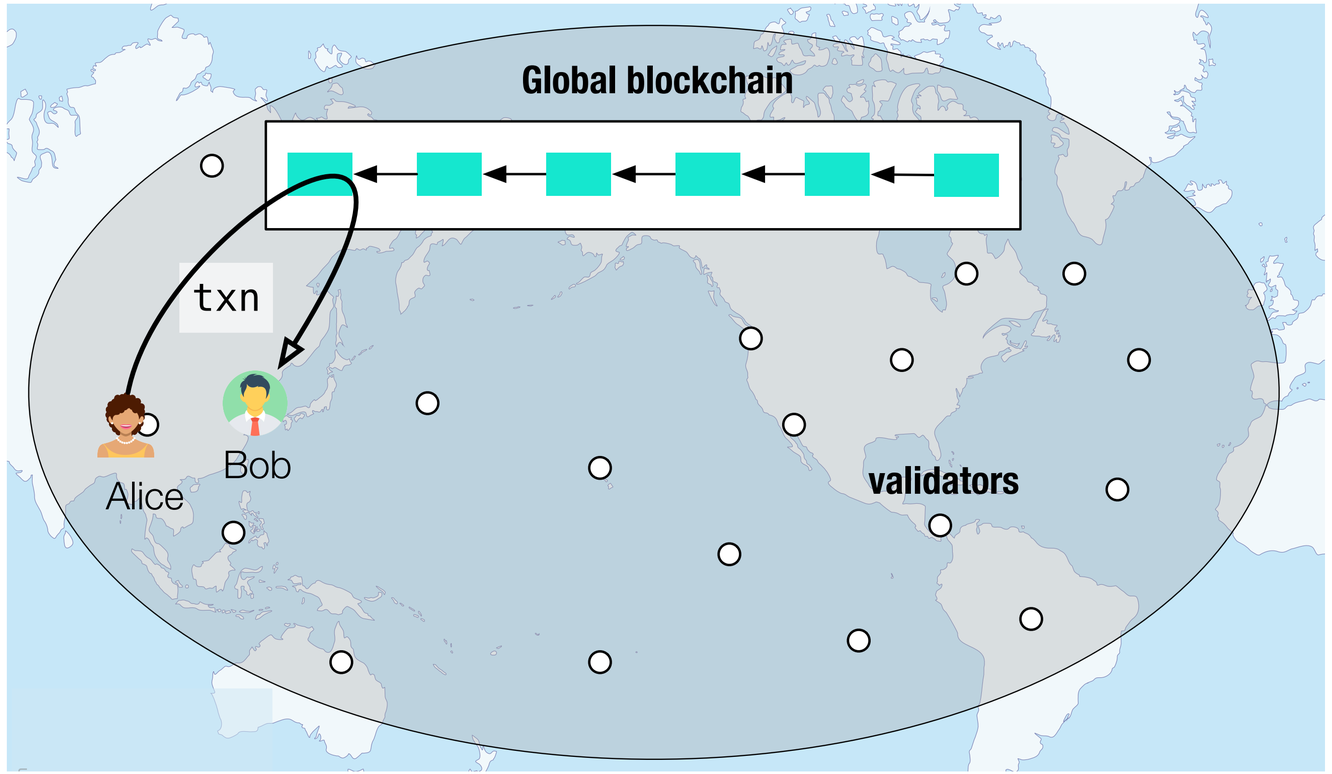




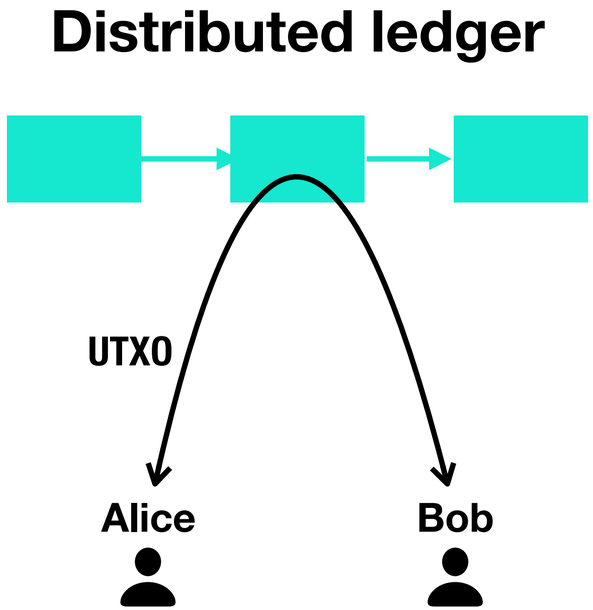
- **Many payments are localized**

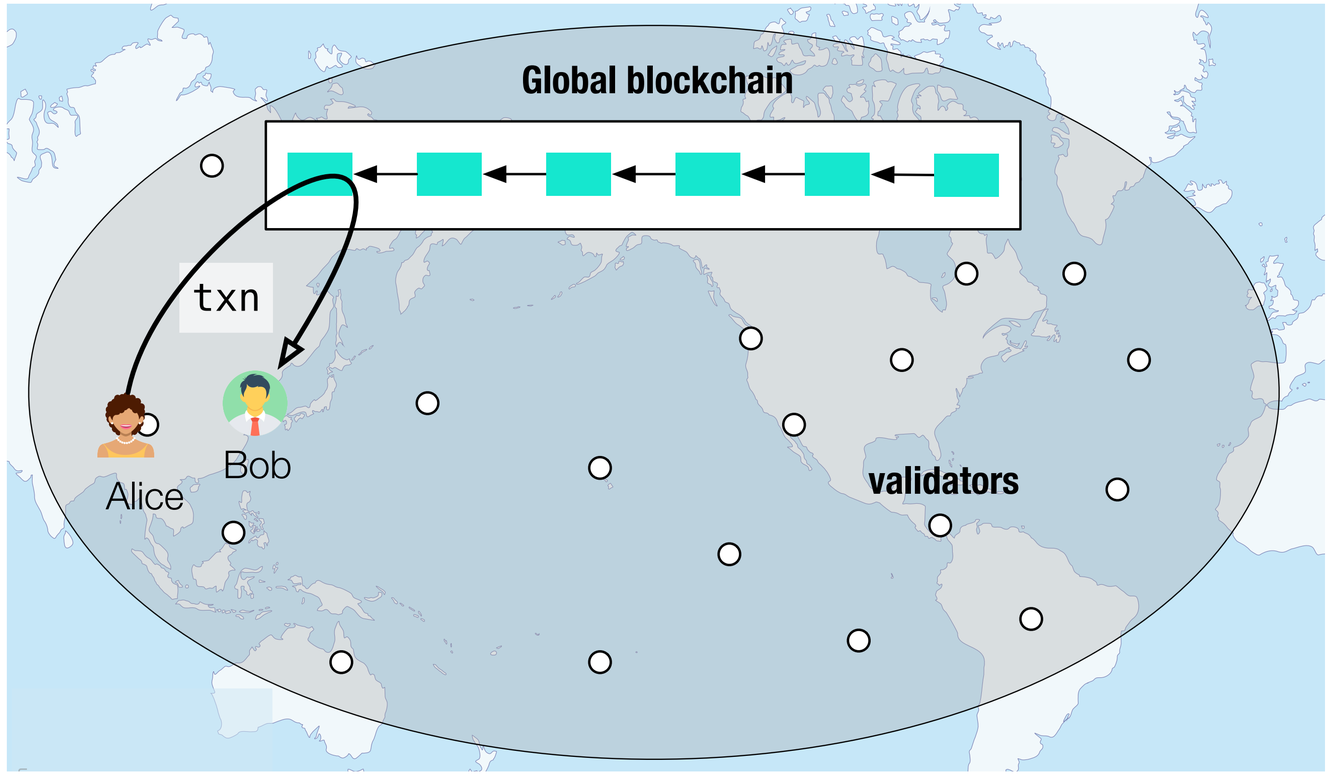
Distributed ledger





- **Many payments are localized**
- **Do we need round-the-world consensus for local payments?**





- **Many payments are localized**
- **Do we need round-the-world consensus for local payments?**
- **Can we enforce local legistation provided KYC is in place?**

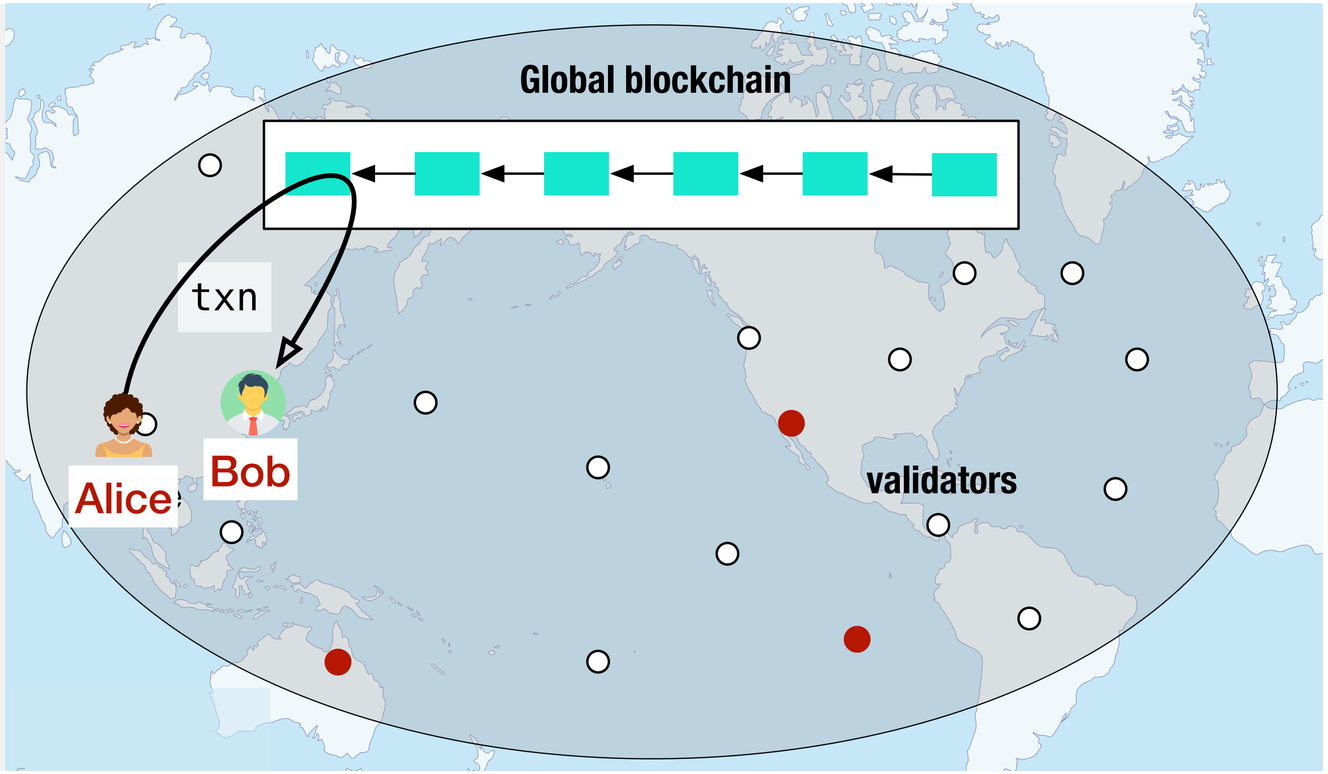
Setup and goals

Setup and goals

- **Same availability goal as Limix, but …**

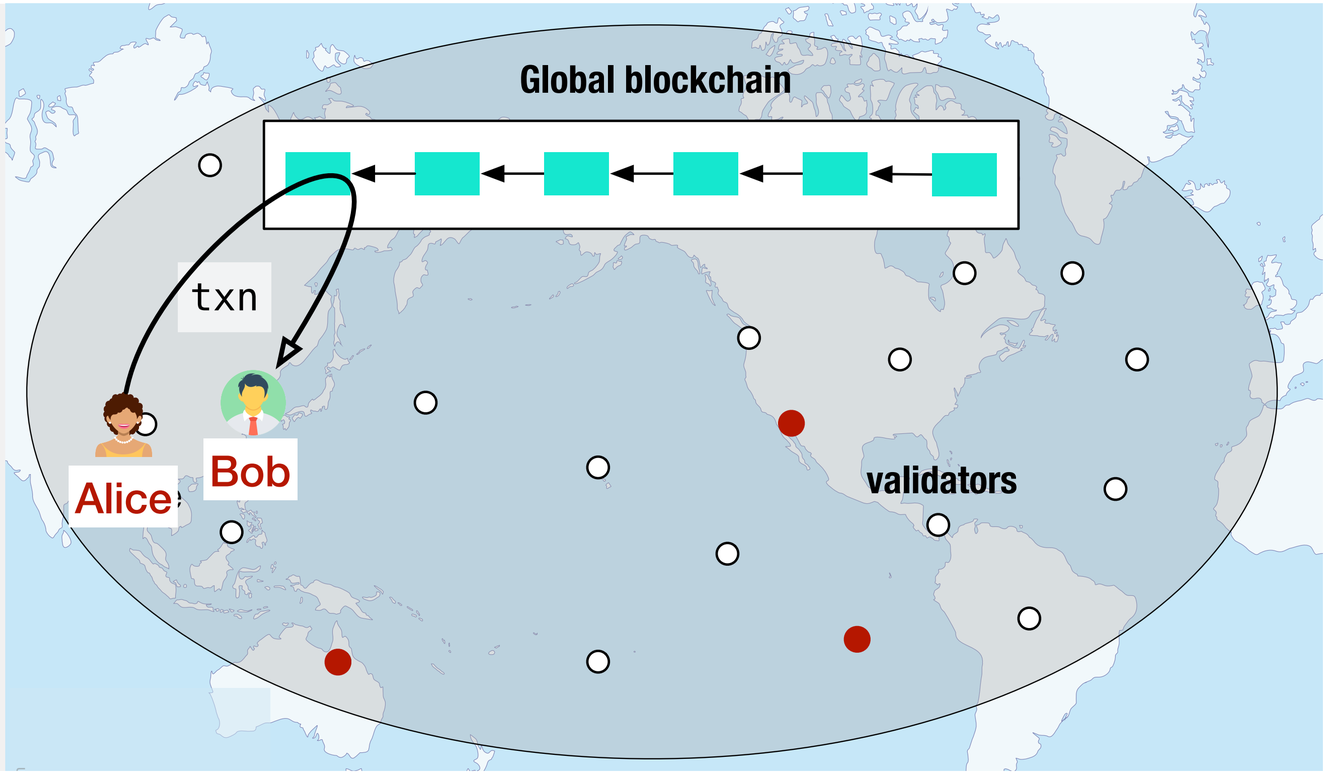
Setup and goals

- **Same availability goal as Limix, but …**



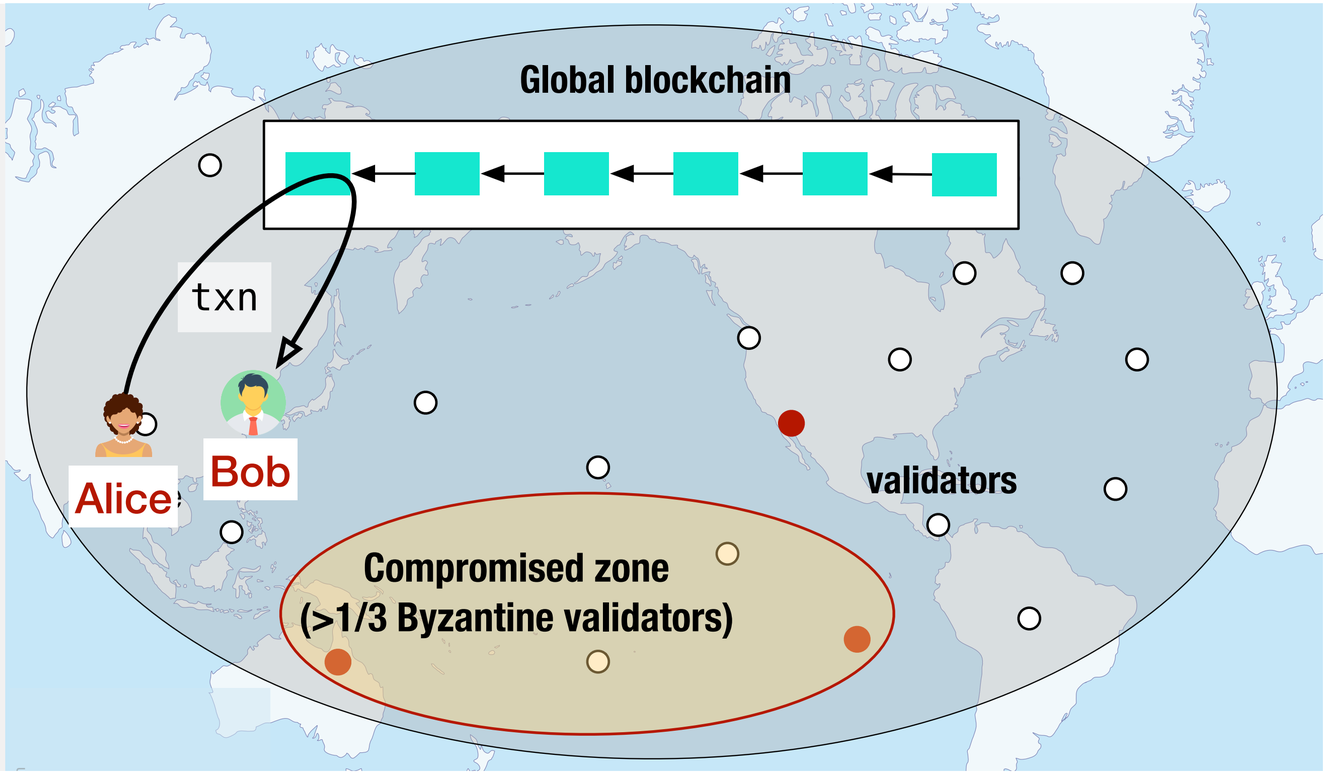
Setup and goals

- **Same availability goal as Limix, but …**
- **Users and validators (< 1/3 globally) can be Byzantine**



Setup and goals

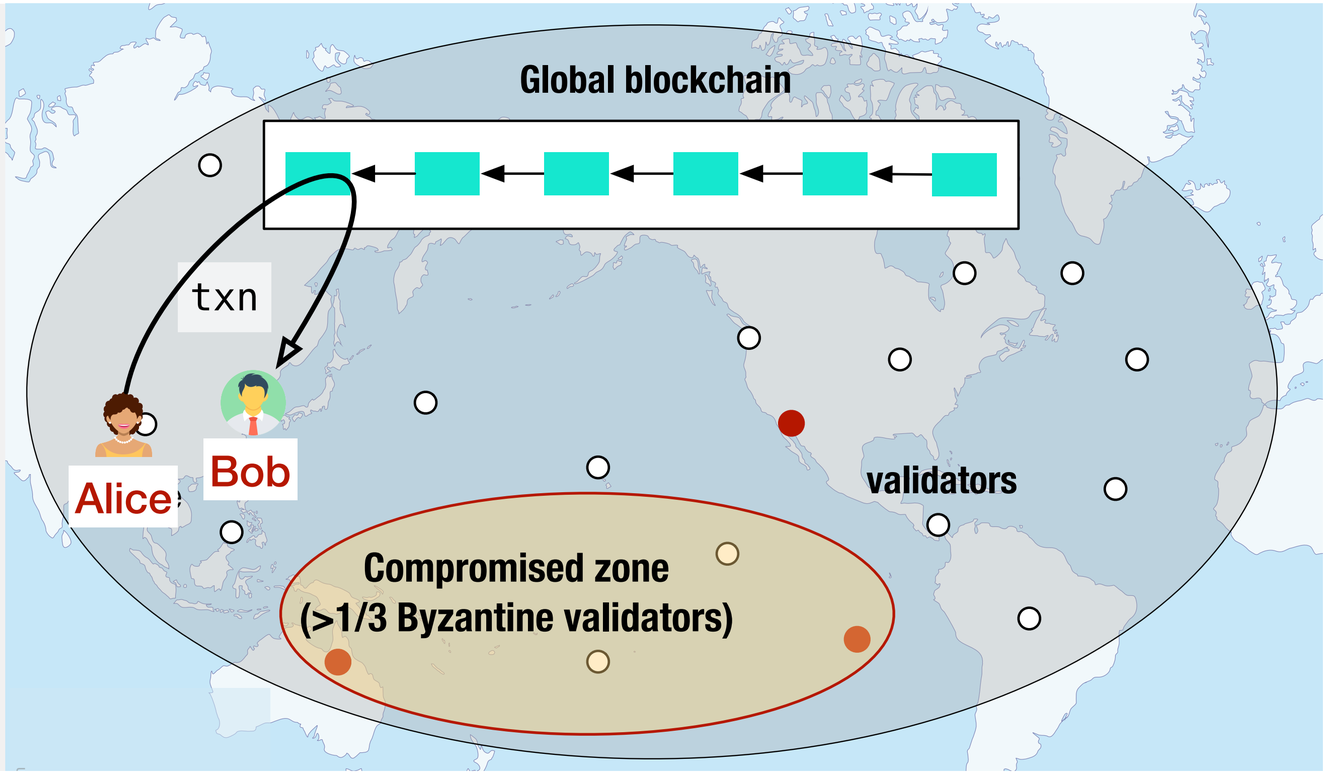
- **Same availability goal as Limix, but …**
- **Users and validators (< 1/3 globally) can be Byzantine**



Setup and goals

- **Same availability goal as Limix, but …**
- **Users and validators (< 1/3 globally) can be Byzantine**

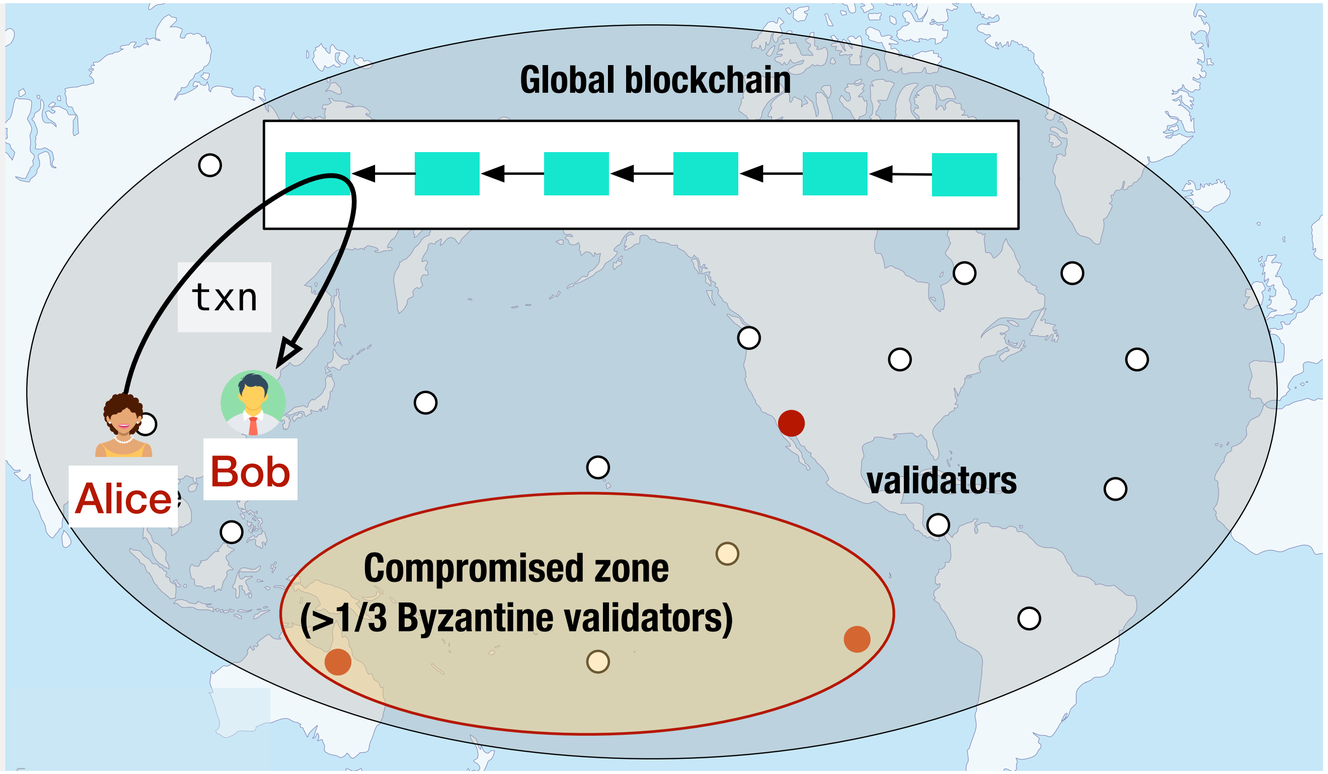
Client attacks	Validator attacks
Double spend	Approve double spend
Zone overload	Censor clients
	Craft its location



Setup and goals

- **Same availability goal as Limix, but …**
- **Users and validators (< 1/3 globally) can be Byzantine**

Client attacks	Validator attacks
Double spend	Approve double spend
Zone overload	Censor clients
	Craft its location

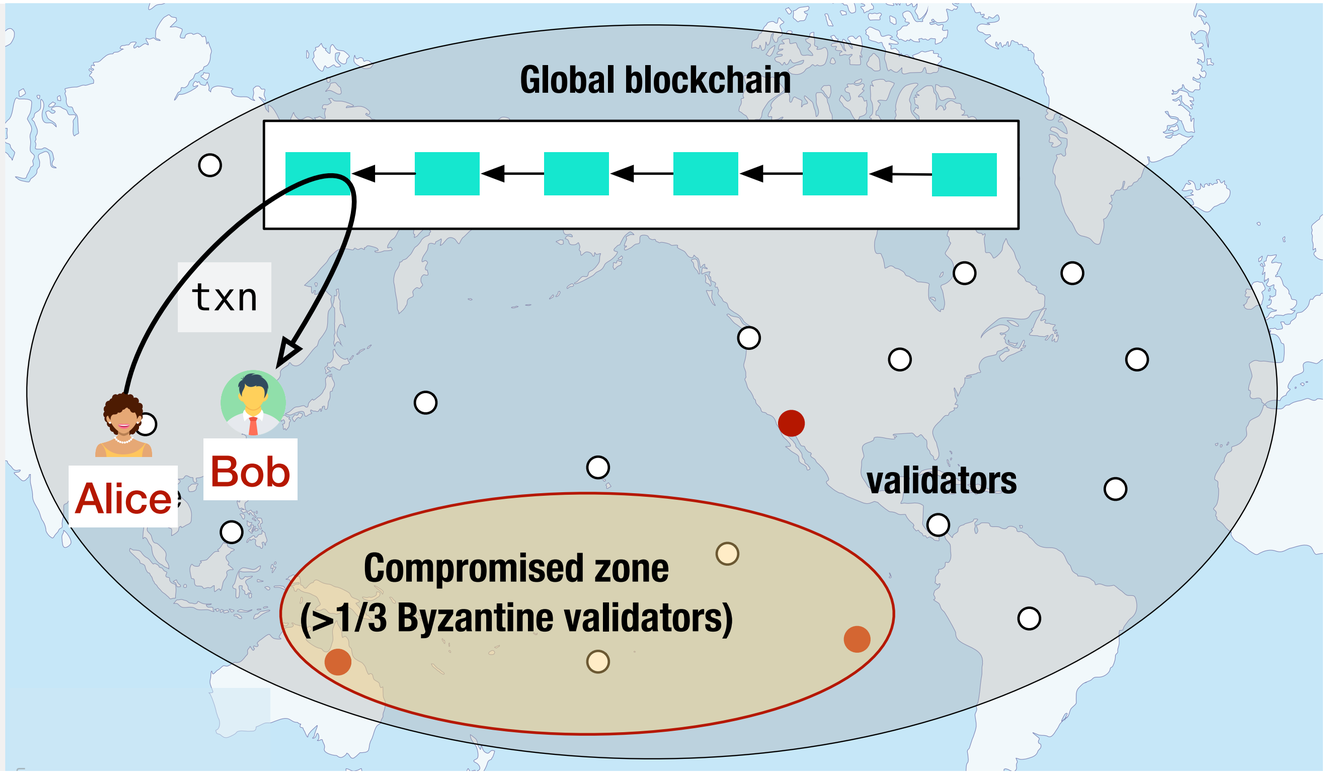


Setup and goals

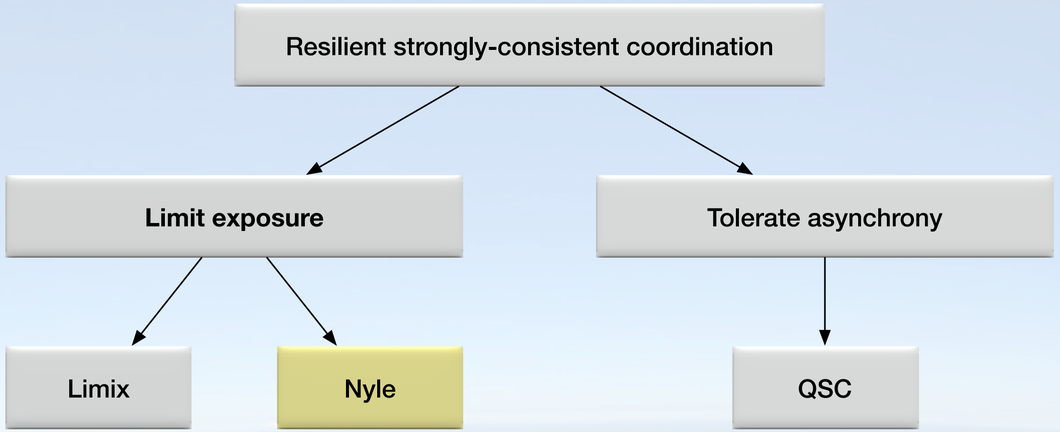
- **Same availability goal as Limix, but …**
- **Users and validators (< 1/3 globally) can be Byzantine**

Client attacks	Validator attacks
Double spend	Approve double spend
Zone overload	Censor clients
	Craft its location

Users decide

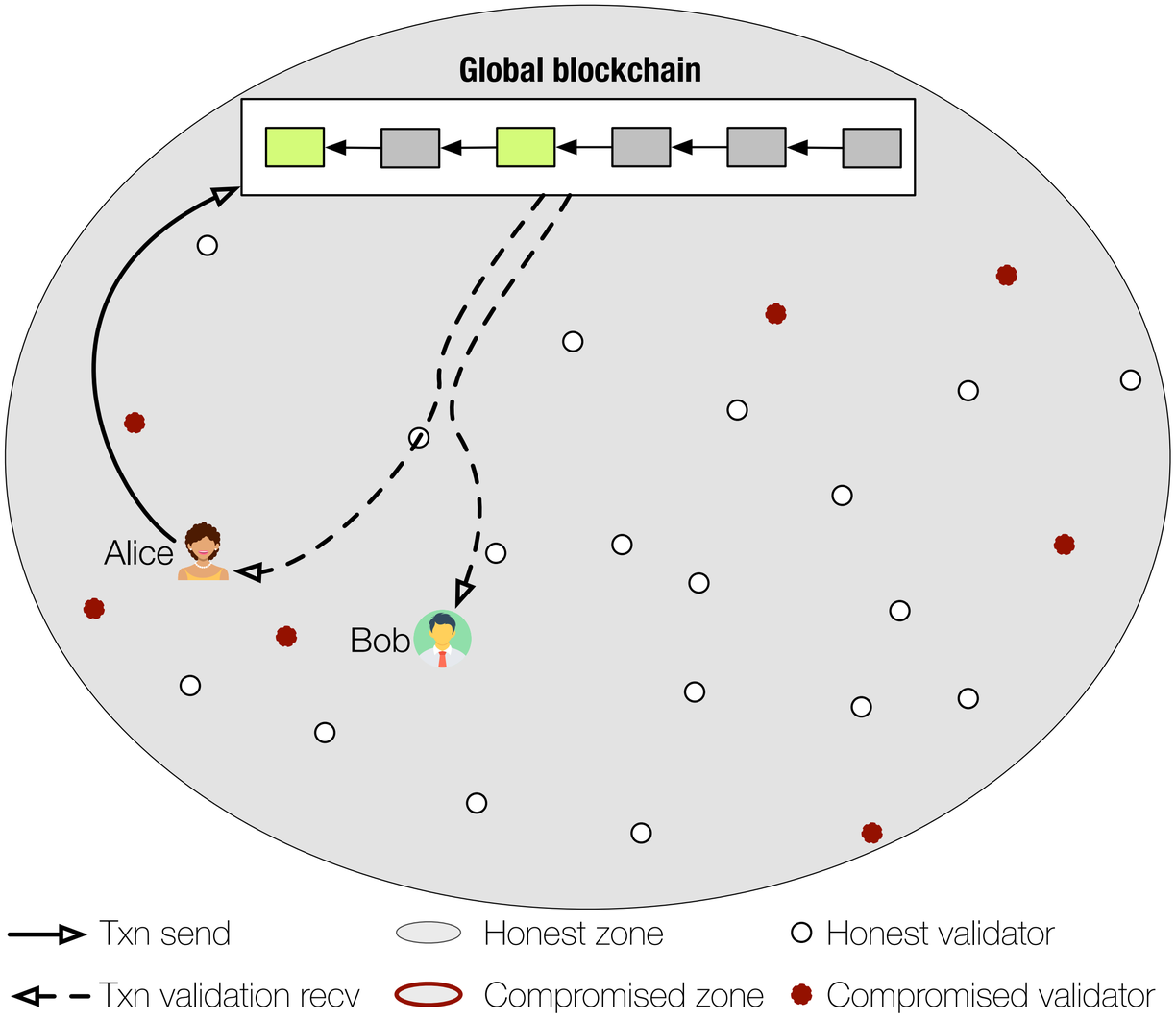


Nyle roadmap



- The problem: exposure-limiting distributed ledger
- Secure zoning with Byzantine validators
- **A trust-but-verify design for transaction validation**
  - Exposing double-spending
  - Preventing zone overload
  - Joint sender-receiver control over exposure
- Multi-input multi-output transactions
- Clearance zone change and censorship attacks
- Simulation
- Summary

Trust-but-verify design

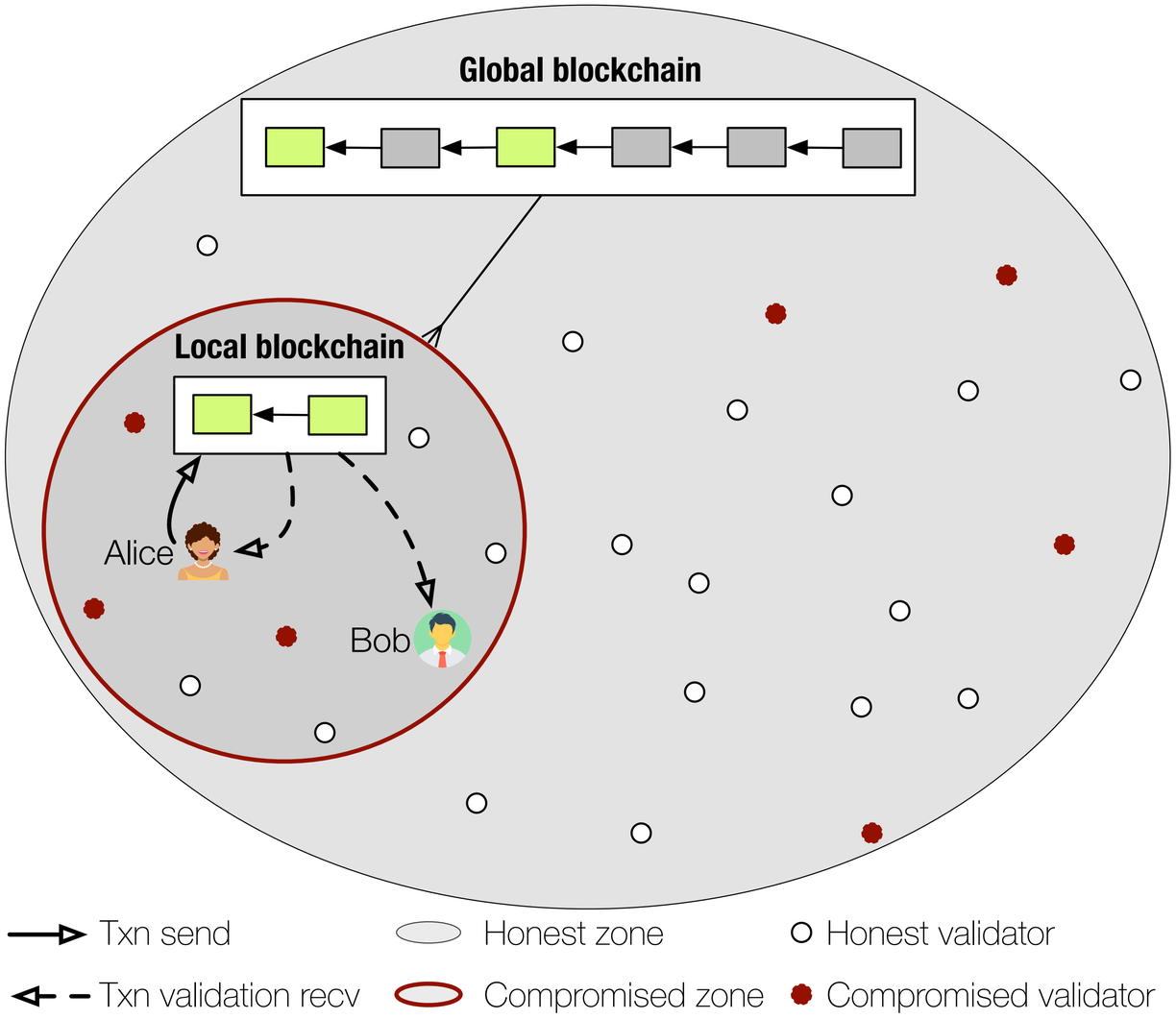


Global blockchain

Prevents double spending

No local availability

Trust-but-verify design



Global blockchain

Prevents double spending

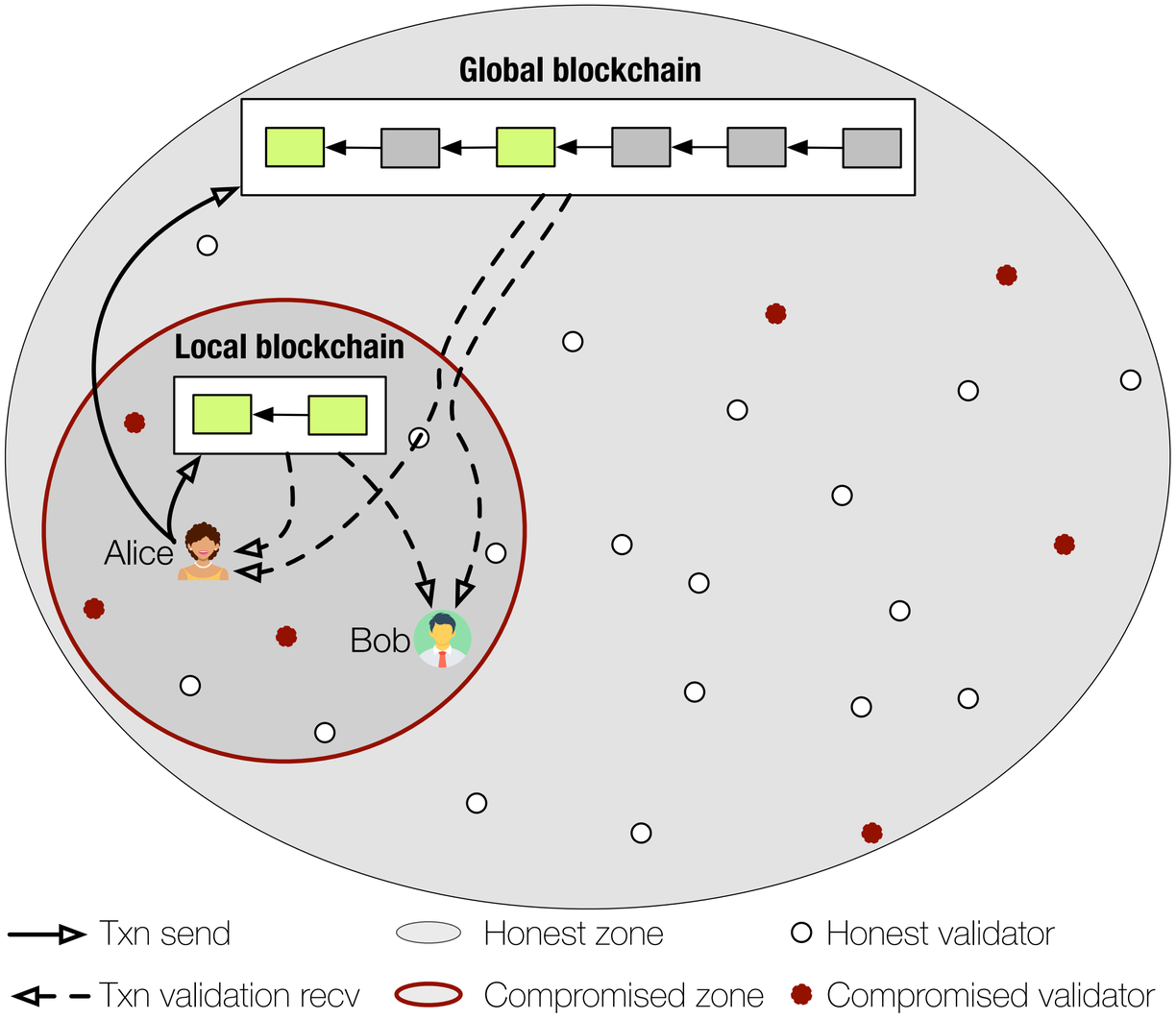
No local availability

Local blockchain validations

Local availability if local zone honest

Censorship and double spending if local zone compromised

Trust-but-verify design



Global blockchain

Prevents double spending

No local availability

Local blockchain validations

Local availability if local zone honest

Censorship and double spending if local zone compromised

Trust-but-verify

Local availability if local zone honest

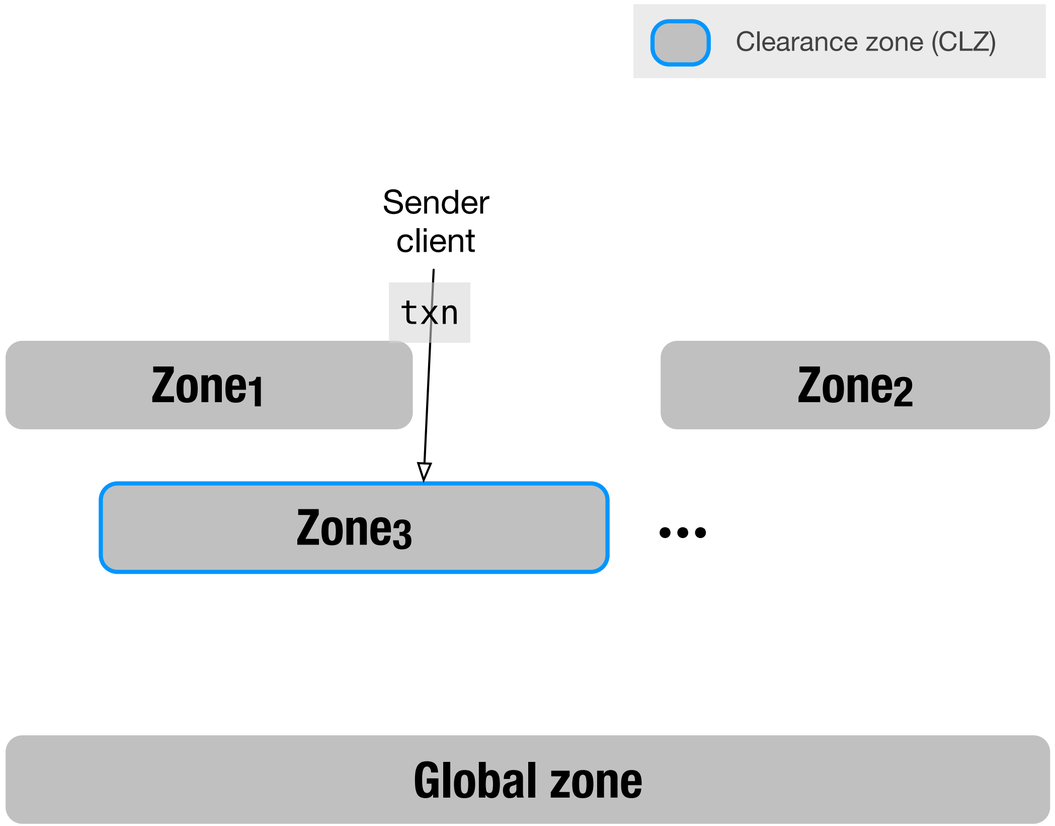
Retroactively detect local double spending

User's choice

Transaction validation

Challenge 1: Sender S double spends via more zones

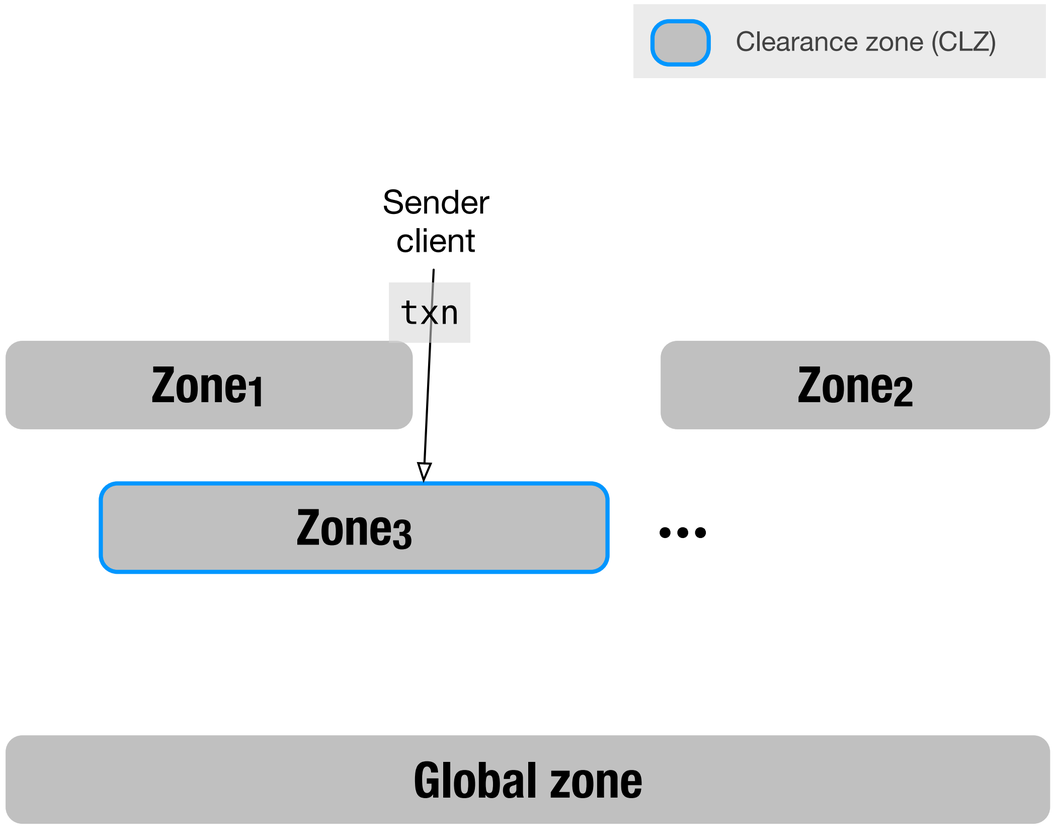
Transaction validation



Challenge 1: Sender S double spends via more zones

Solution: One clearance zone (CLZ) per UTXO approves spend

Transaction validation

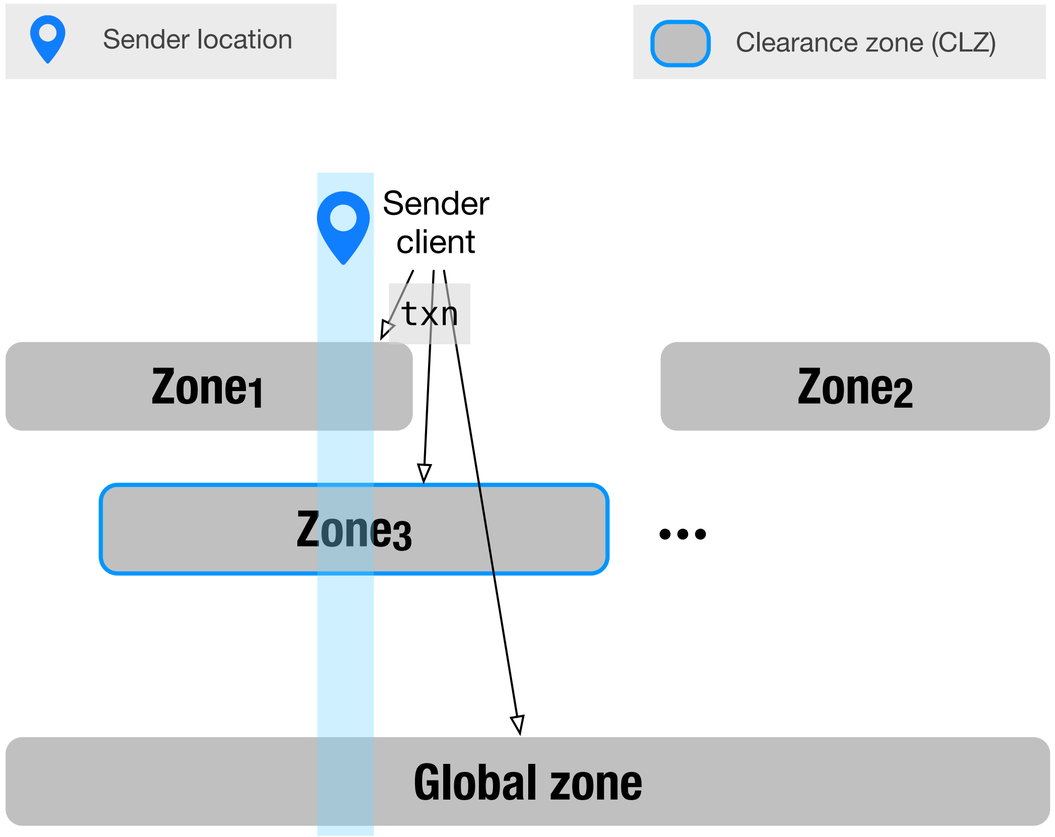


Challenge 1: Sender S double spends via more zones

Solution: One clearance zone (CLZ) per UTXO approves spend

Challenge 2: Sender S overloads zones

Transaction validation



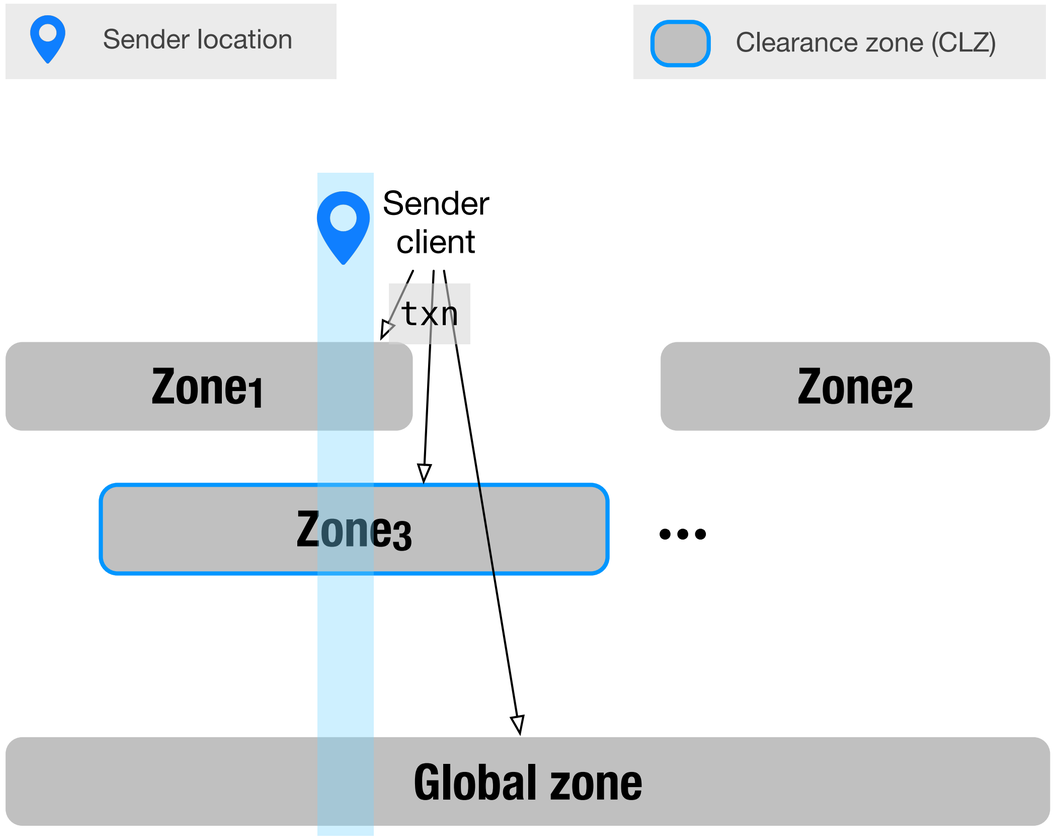
Challenge 1: Sender S double spends via more zones

Solution: One clearance zone (CLZ) per UTXO approves spend

Challenge 2: Sender S overloads zones

Solution: S declares location, sends txn only to local zones

Transaction validation



Challenge 1: Sender S double spends via more zones

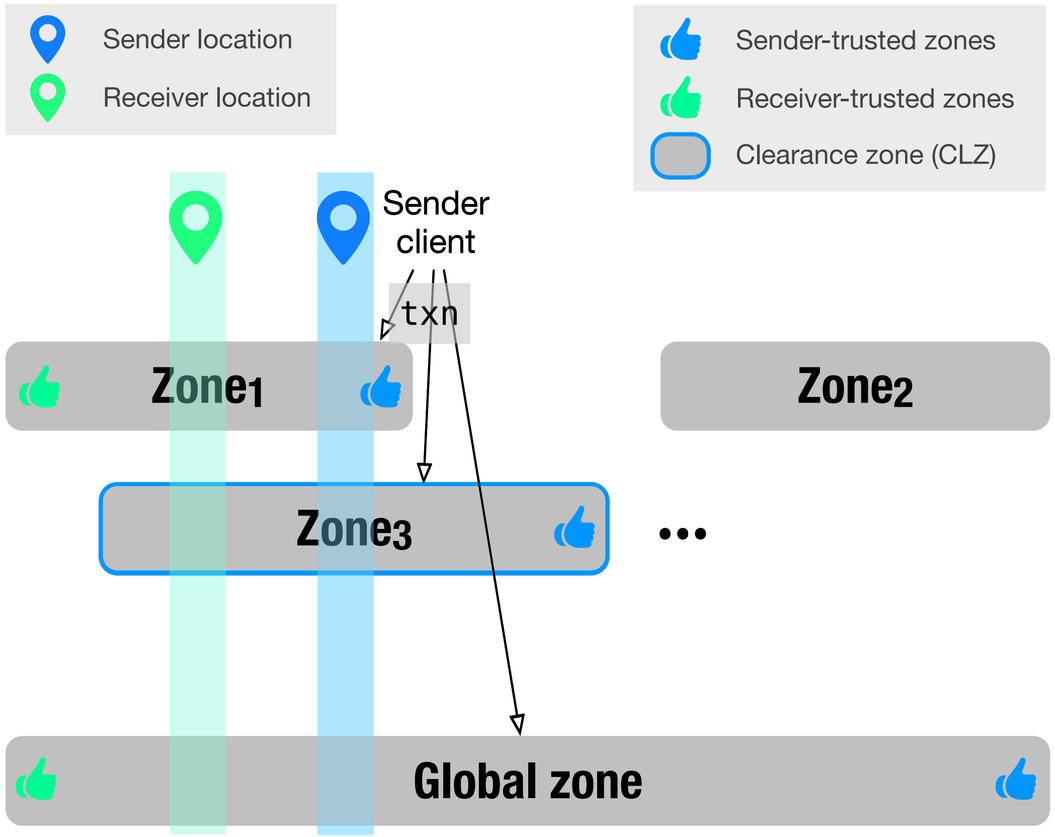
Solution: One clearance zone (CLZ) per UTXO approves spend

Challenge 2: Sender S overloads zones

Solution: S declares location, sends txn only to local zones

Challenge 3: Joint sender-receiver control over exposure

Transaction validation



Challenge 1: Sender S double spends via more zones

Solution: One clearance zone (CLZ) per UTXO approves spend

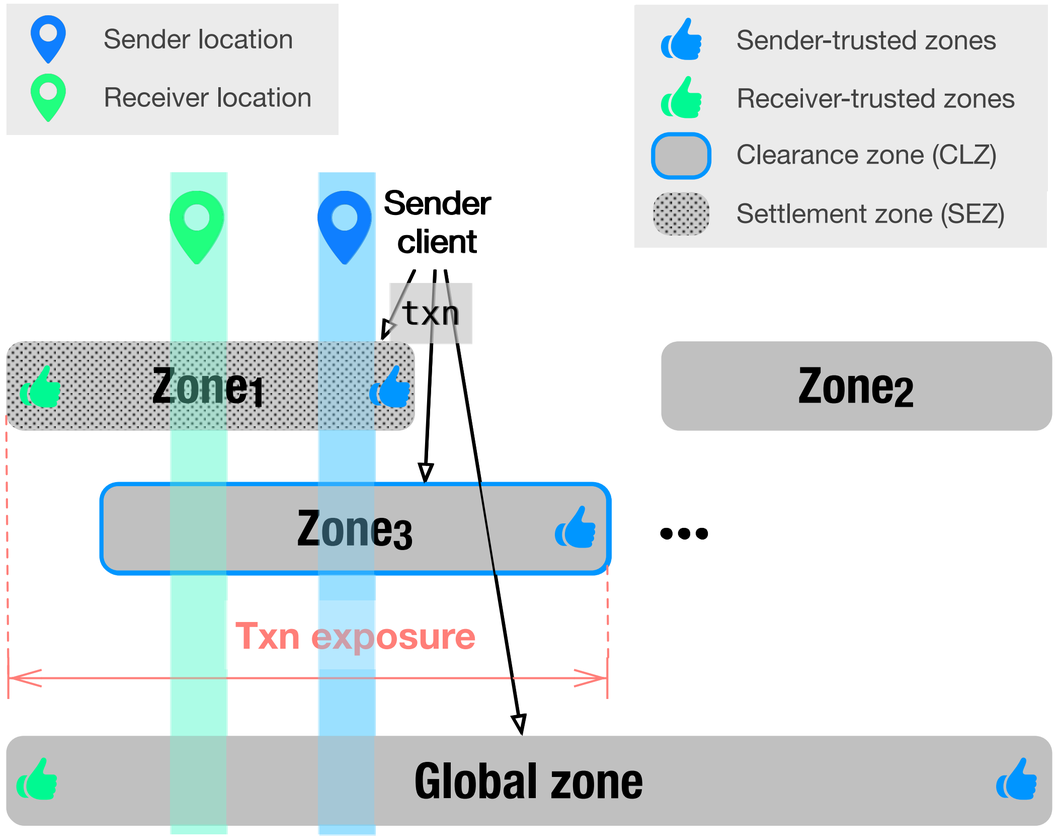
Challenge 2: Sender S overloads zones

Solution: S declares location, sends txn only to local zones

Challenge 3: Joint sender-receiver control over exposure

Solution: S, R specify trusted zones => settlement zone (SEZ)

Transaction validation



Challenge 1: Sender S double spends via more zones

Solution: One clearance zone (CLZ) per UTXO approves spend

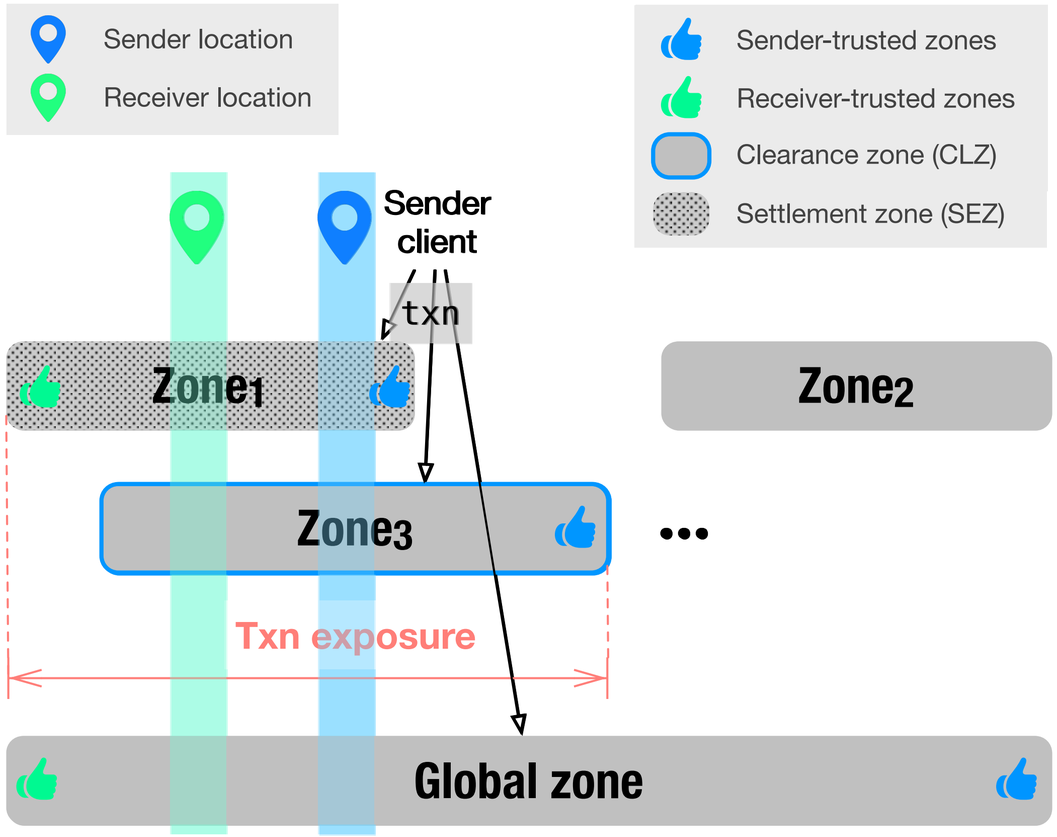
Challenge 2: Sender S overloads zones

Solution: S declares location, sends txn only to local zones

Challenge 3: Joint sender-receiver control over exposure

Solution: S, R specify trusted zones => settlement zone (SEZ)

Transaction validation



Challenge 1: Sender S double spends via more zones

Solution: One clearance zone (CLZ) per UTXO approves spend

Challenge 2: Sender S overloads zones

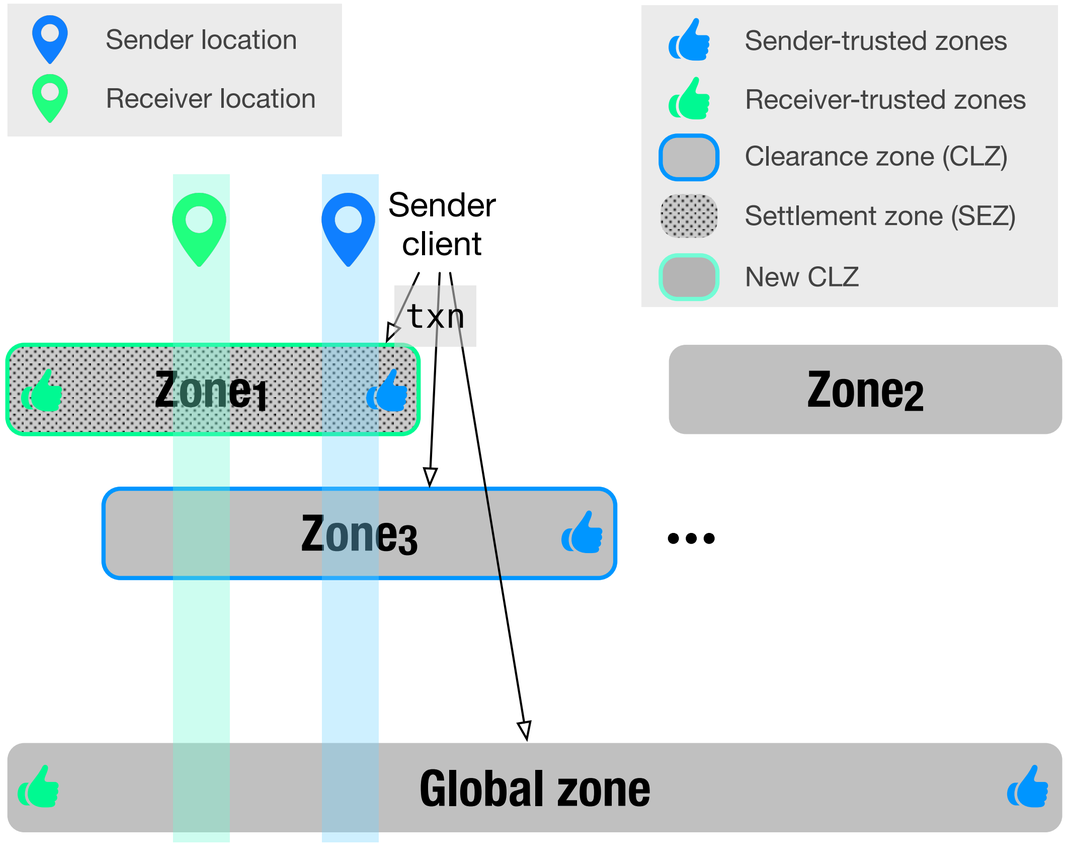
Solution: S declares location, sends txn only to local zones

Challenge 3: Joint sender-receiver control over exposure

Solution: S, R specify trusted zones => settlement zone (SEZ)

Challenge 4: CLZ trusted by receiver (R) for new UTXO

Transaction validation



Challenge 1: Sender S double spends via more zones

Solution: One clearance zone (CLZ) per UTXO approves spend

Challenge 2: Sender S overloads zones

Solution: S declares location, sends txn only to local zones

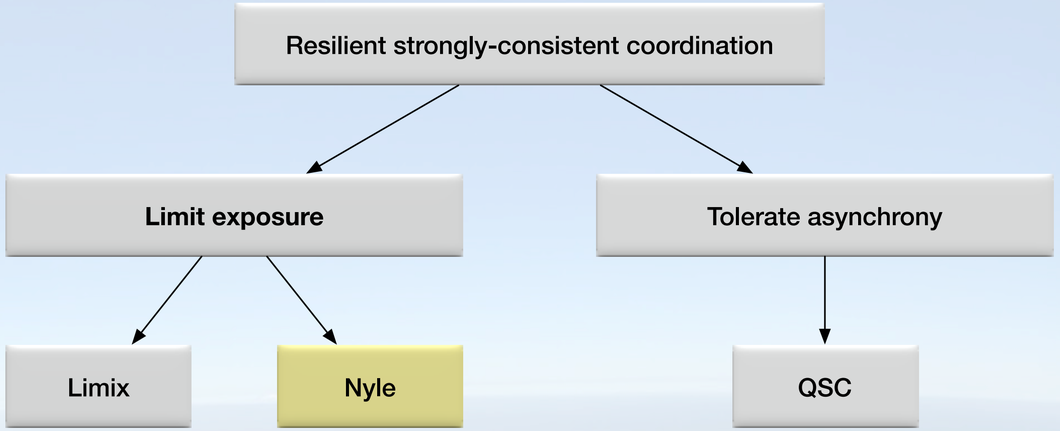
Challenge 3: Joint sender-receiver control over exposure

Solution: S, R specify trusted zones => settlement zone (SEZ)

Challenge 4: CLZ trusted by receiver (R) for new UTXO

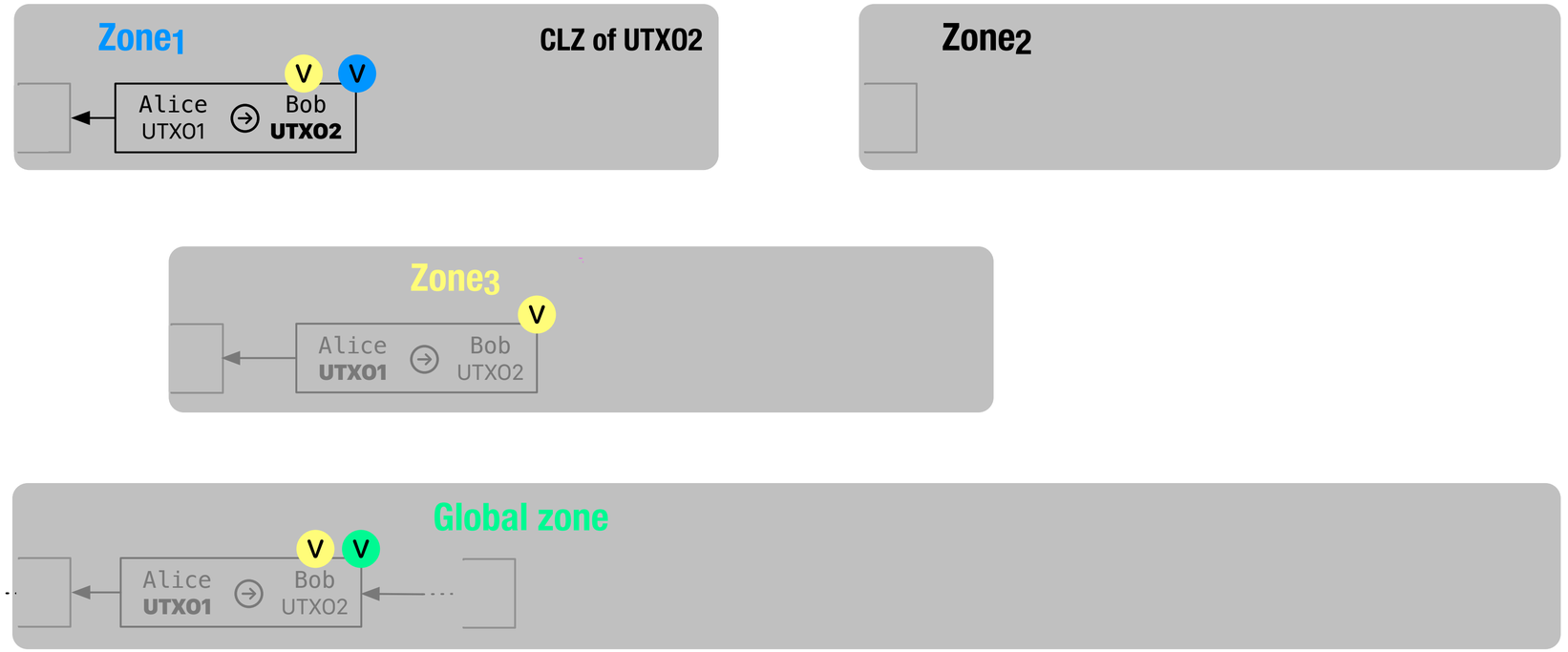
Solution: CLZ is smallest R trusted zone that validates txn

Nyle roadmap



- The problem: exposure-limiting distributed ledger
- Secure zoning with Byzantine validators
- A trust-but-verify design for transaction validation
- Multi-input multi-output transactions
- **Clearance zone change and censorship attacks**
- Simulation
- Summary

Changing the CLZ



V V V Transaction validation by a zone. The color identifies the zone.

Pending transaction validation by a zone. The color identifies the zone.

Zone transfer txn

Token transfer txn

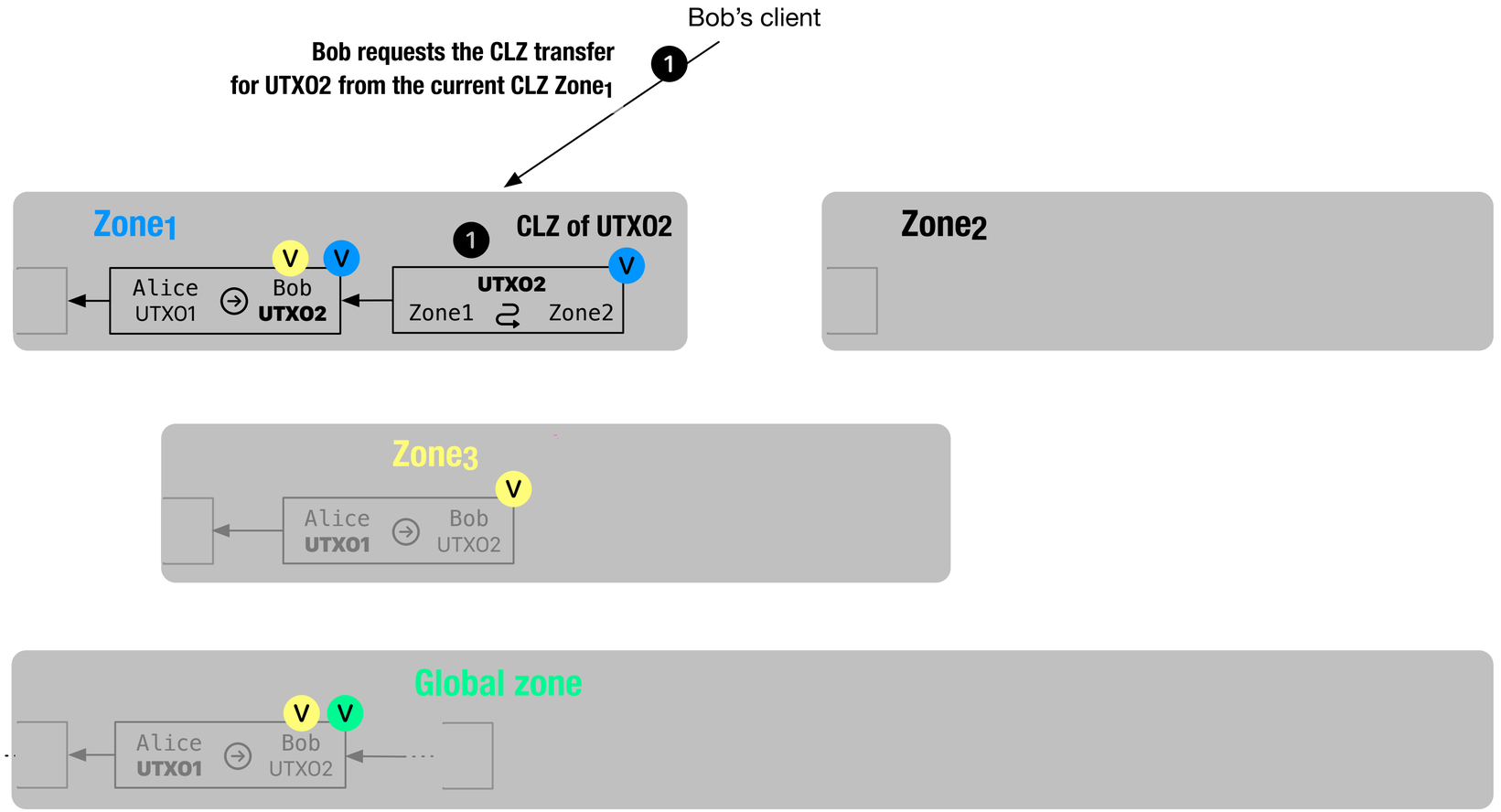
Use cases

User changes location

User does not trust zone anymore, e.g., after proven misbehavior

User transfers funds to a single zone for a multi-input txn

Changing the CLZ



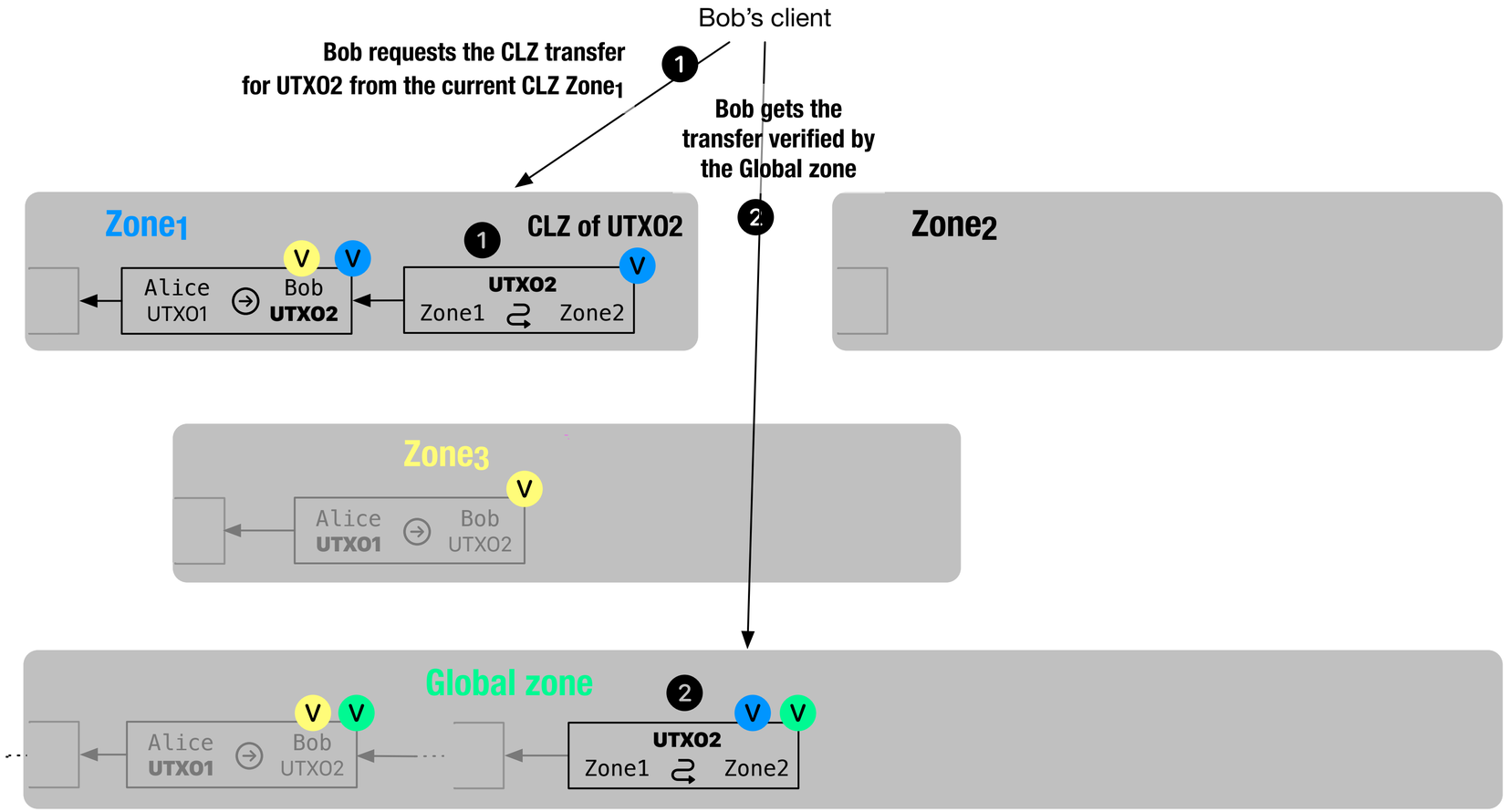
Use cases

User changes location

User does not trust zone anymore, e.g., after proven misbehavior

User transfers funds to a single zone for a multi-input txn

Changing the CLZ



Transaction validation by a zone. The color identifies the zone.

Pending transaction validation by a zone. The color identifies the zone.

Zone transfer txn

Token transfer txn

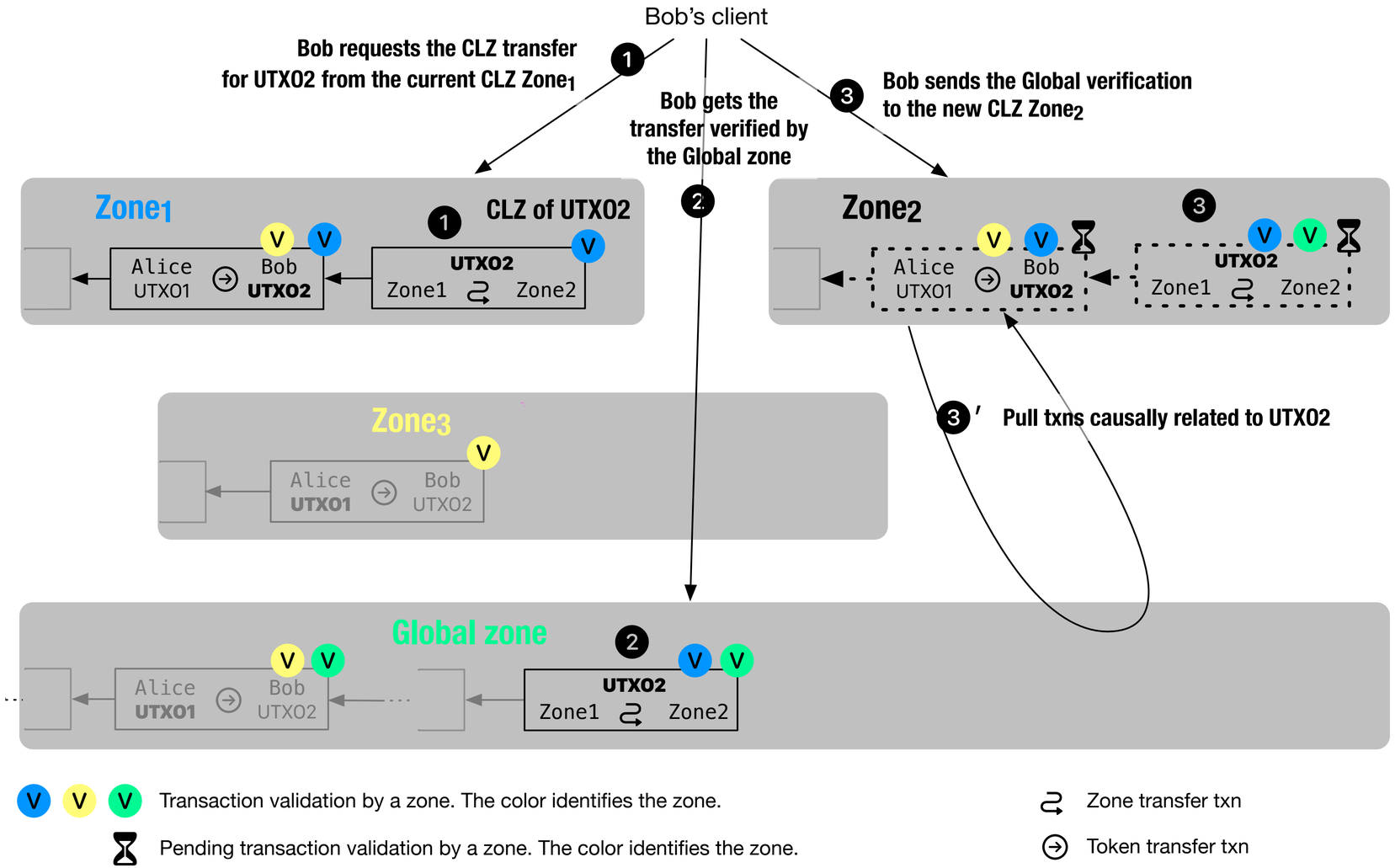
Use cases

User changes location

User does not trust zone anymore, e.g., after proven misbehavior

User transfers funds to a single zone for a multi-input txn

Changing the CLZ



Use cases

User changes location

User does not trust zone anymore, e.g., after proven misbehavior

User transfers funds to a single zone for a multi-input txn

Defending against CLZ censorship

Defending against CLZ censorship

- **CLZ censors the UTXO owner**

Defending against CLZ censorship

- **CLZ censors the UTXO owner**
  - User complains to Global zone

Defending against CLZ censorship

- **CLZ censors the UTXO owner**
  - User complains to Global zone
  - Global zone performs the CLZ transfer protocol in user's name

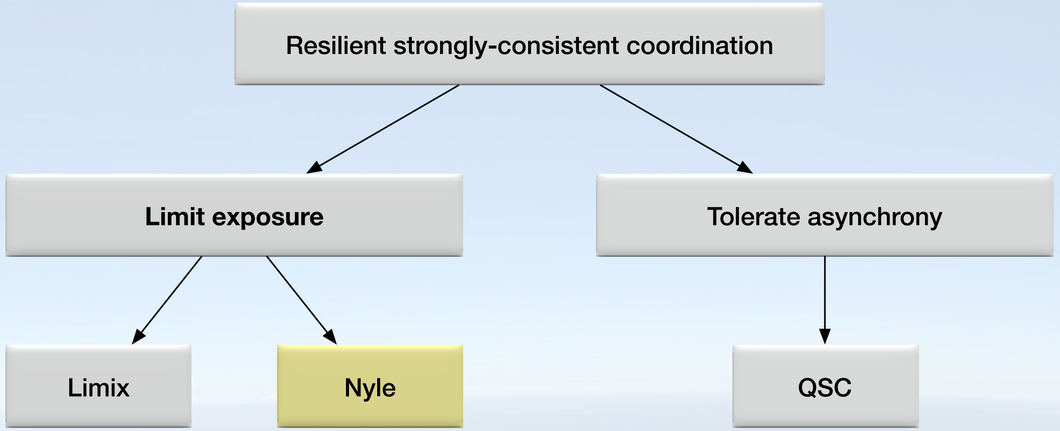
Defending against CLZ censorship

- **CLZ censors the UTXO owner**
  - User complains to Global zone
  - Global zone performs the CLZ transfer protocol in user's name
- **CLZ appears not live to Global zone (long timeout)**

Defending against CLZ censorship

- **CLZ censors the UTXO owner**
  - User complains to Global zone
  - Global zone performs the CLZ transfer protocol in user's name
- **CLZ appears not live to Global zone (long timeout)**
  - Global zone forcibly relinquishes CLZ UTXOs

Nyle roadmap

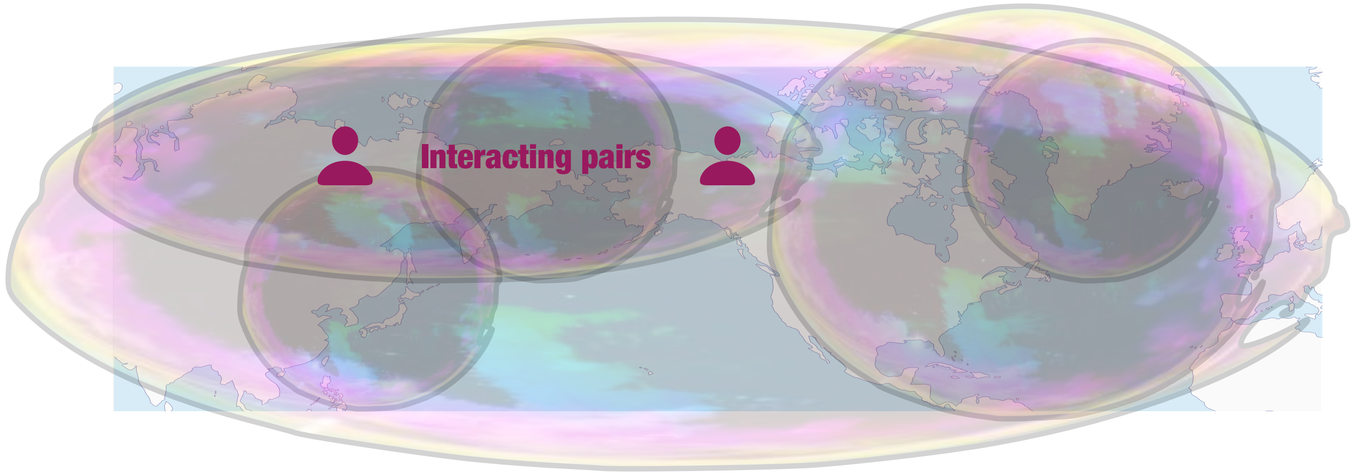


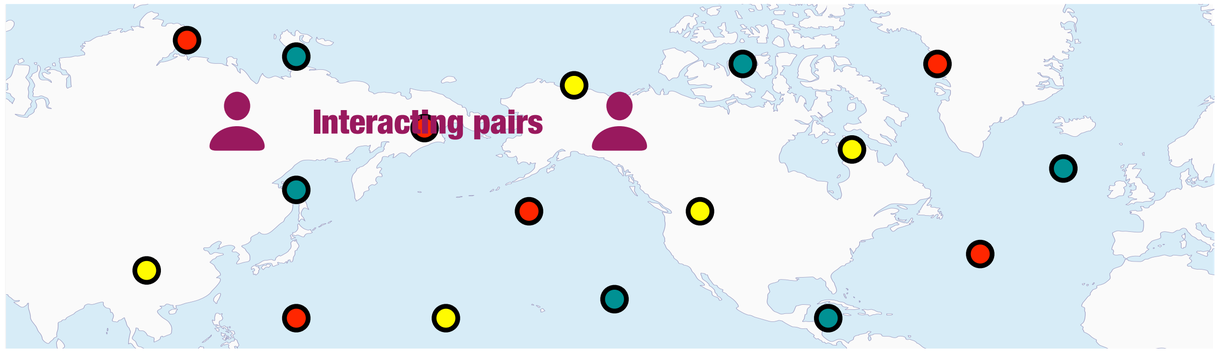
- The problem: exposure-limiting distributed ledger
- Secure zoning with Byzantine validators
- A trust-but-verify design for transaction validation
- Multi-input multi-output transactions
- Clearance zone change and censorship attacks
- **Simulation**
- Summary

Simulation setup

- **Do far away failures affect transactions?**

- 400 nodes, max RTT 1 second





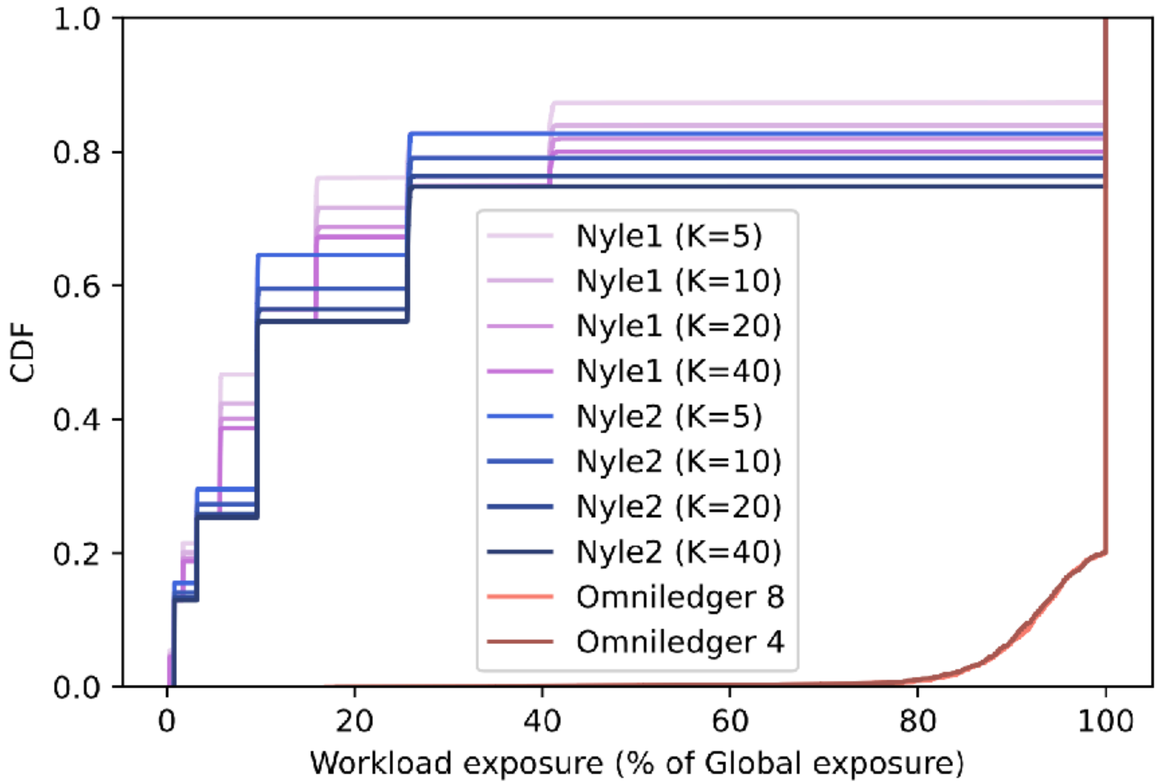
- **Nyle RTT-based autozoning**
  - **Nyle1**: More zones
  - **Nyle2**: Fewer zones

- **Omniledger***
  - **Omniledger4**: Shards of 80 nodes, each with 20 μshards
  - **Omniledger8**: Shards of 80 nodes, each with 10 μshards

* Eleftherios Kokoris-Kogias, Philipp Jovanovic, Linus Gasser, Nicolas Gailly, Ewa Syta, and Bryan Ford. OmniLedger: A Secure, Scale-Out, Decentralized Ledger via Sharding. In *39th IEEE Symposium on Security and Privacy (SP)*, 2018.

Simulation results

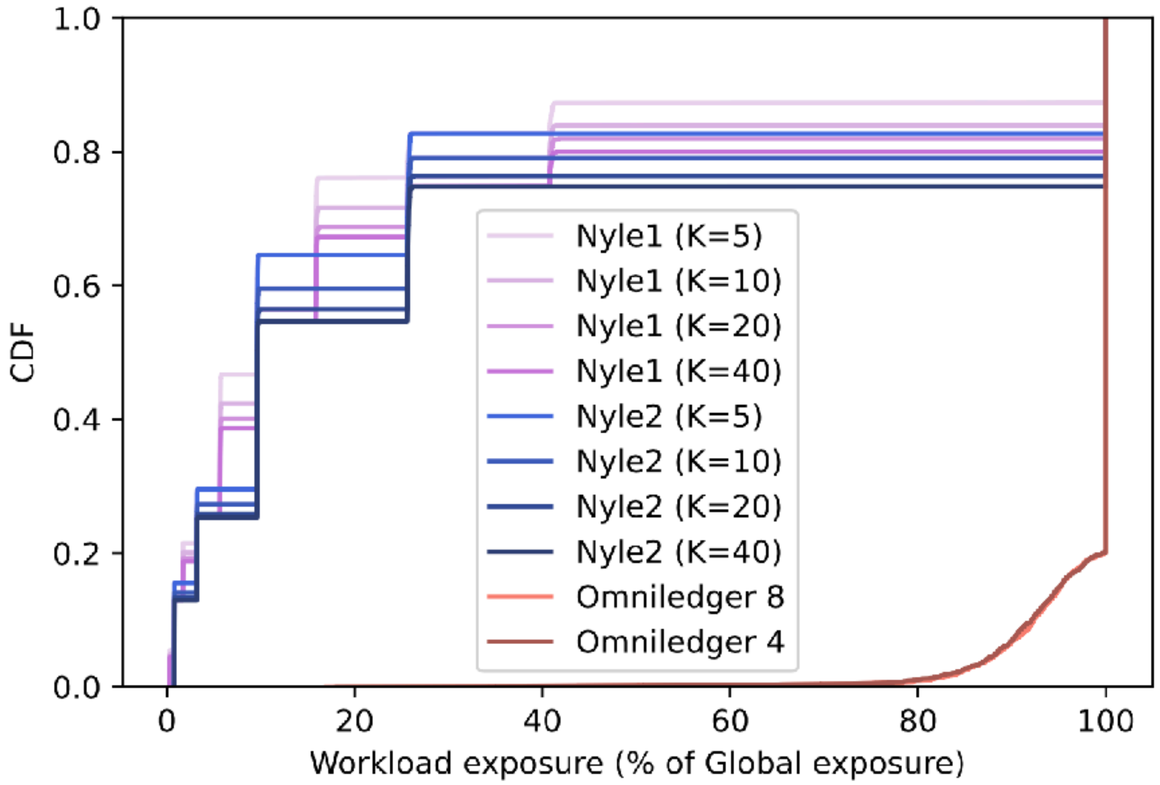
- Workload: 10000 random pairs with RTTs from a realistic trace-based distribution

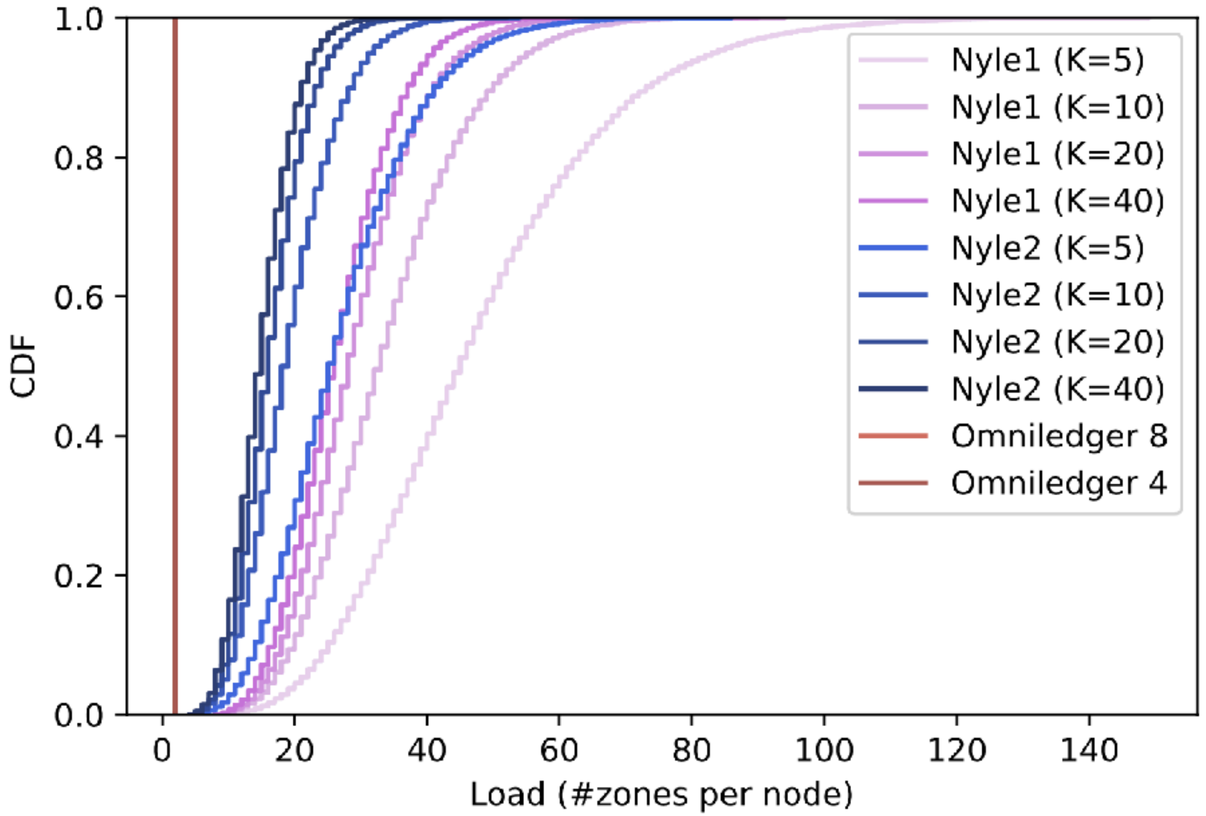


Lower exposure is better, because it guarantees higher availability

Simulation results

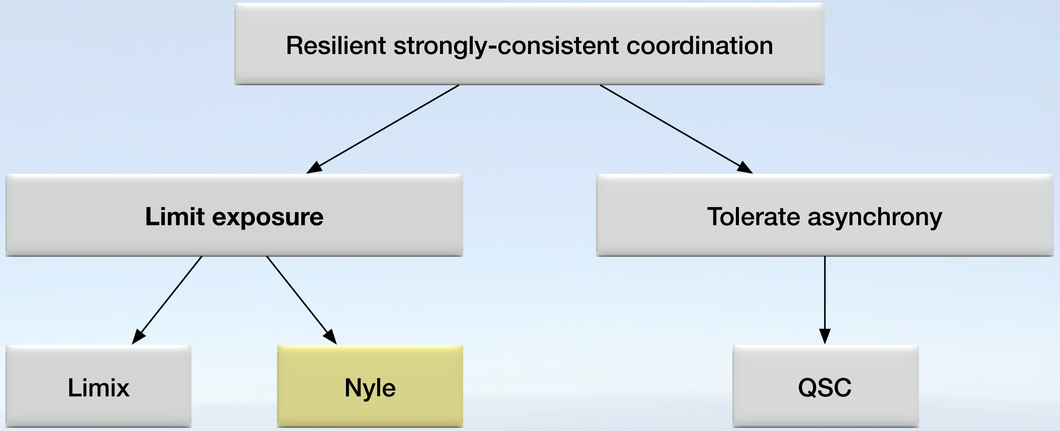
- Workload: 10000 random pairs with RTTs from a realistic trace-based distribution





Lower exposure is better, because it guarantees higher availability

Nyle roadmap

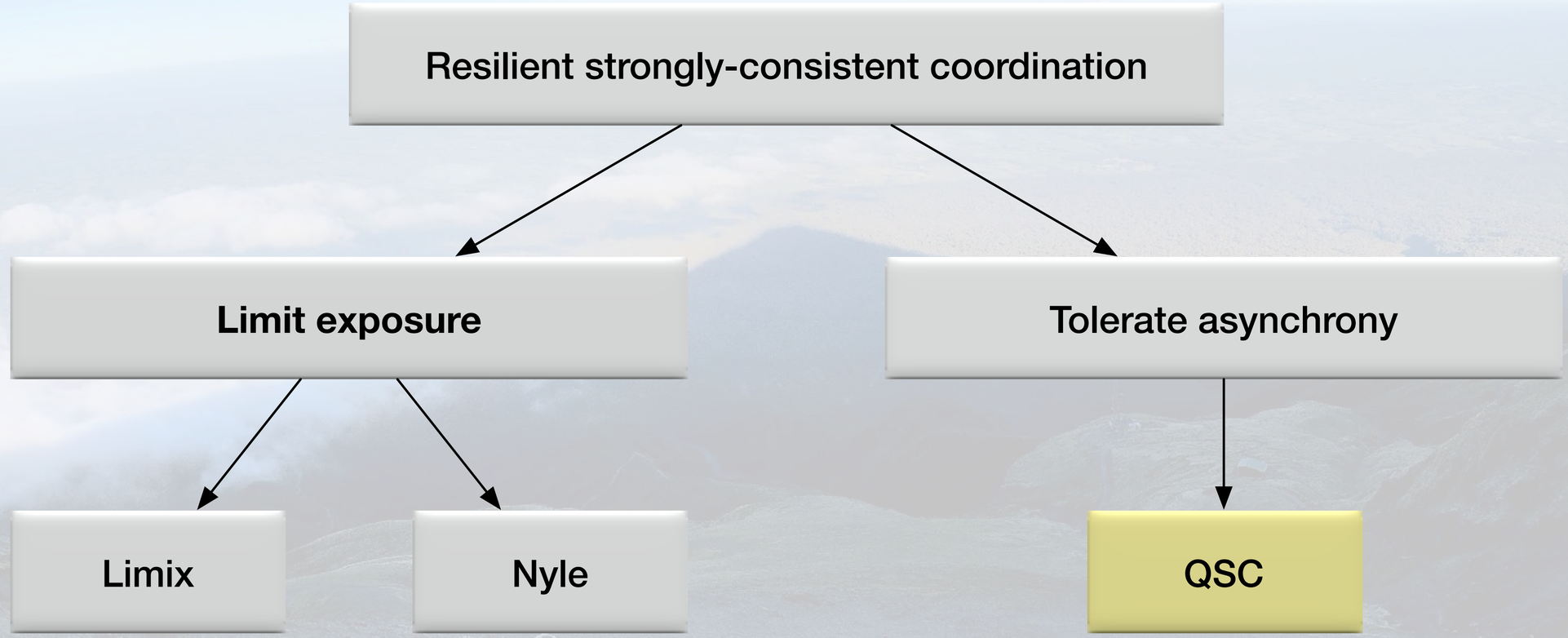


- The problem: exposure-limiting distributed ledger
- Secure zoning with Byzantine validators
- A trust-but-verify design for transaction validation
- Multi-input multi-output transactions
- Clearance zone change and censorship attacks
- Simulation
- **Summary**

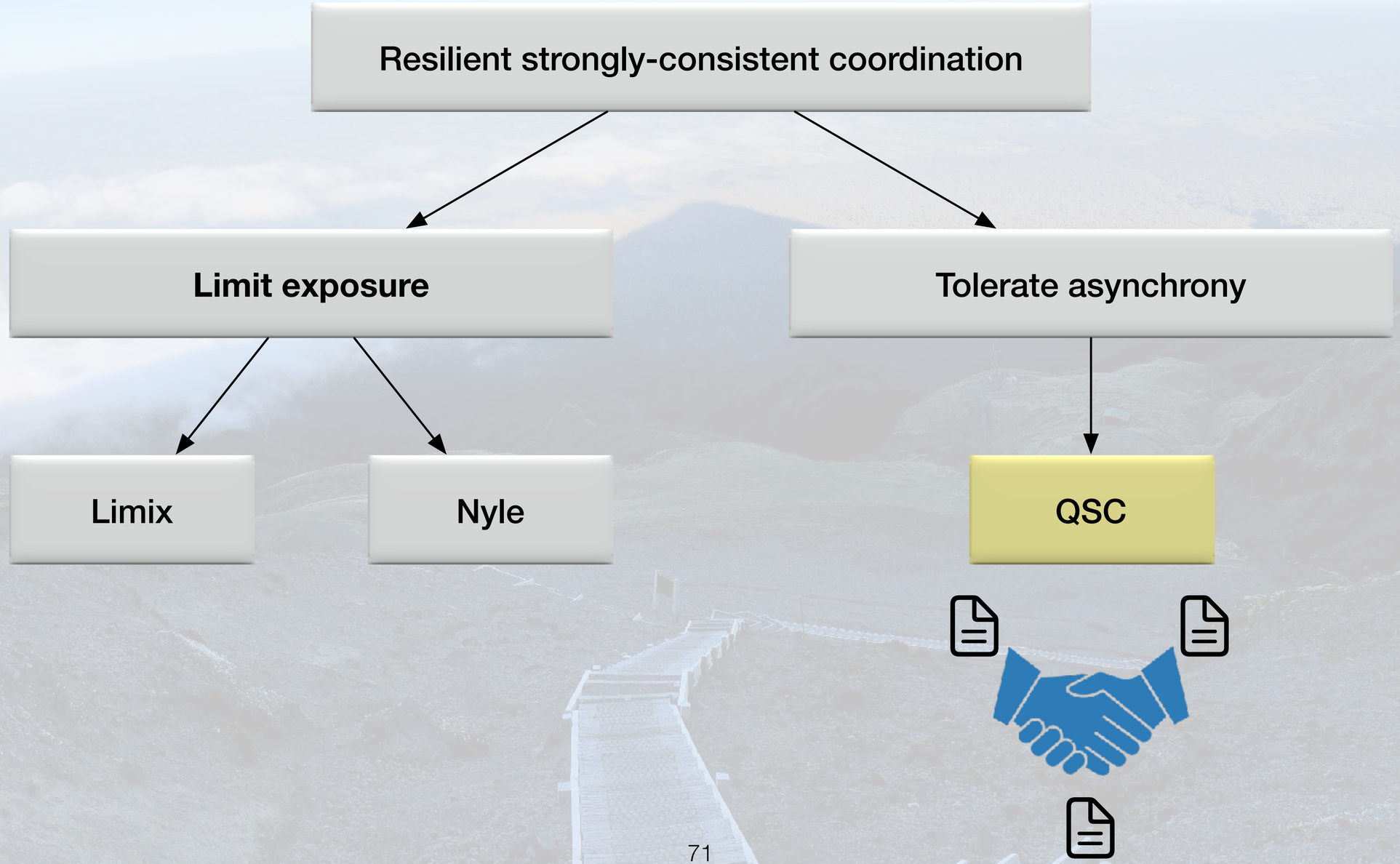
Nyle summary

- **Exposure-limiting architecture for BFT-style distributed ledgers**
  - Global zone is a secure finaliser, whereas local zones are opportunistic finalisers
- **Per-transaction user-controlled exposure**
- **Nyle significantly reduces exposure compared to Omniledger, at a 10x cost**

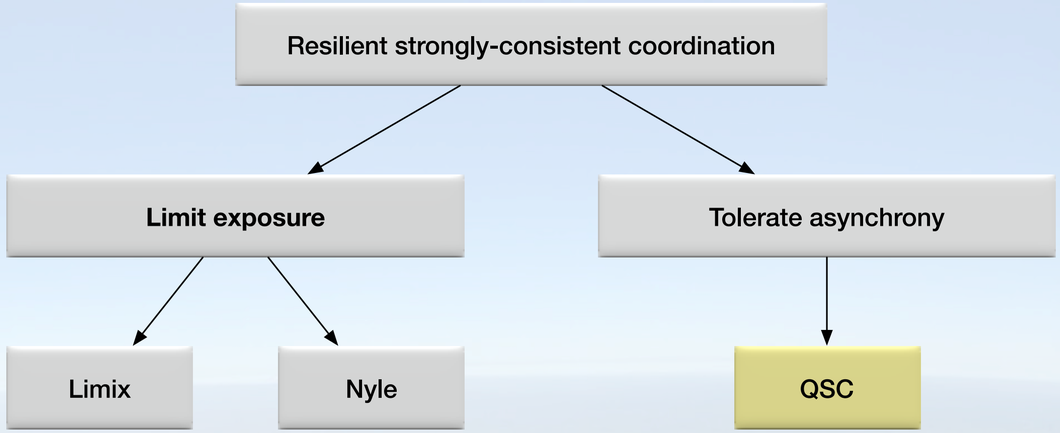
Thesis roadmap



Thesis roadmap

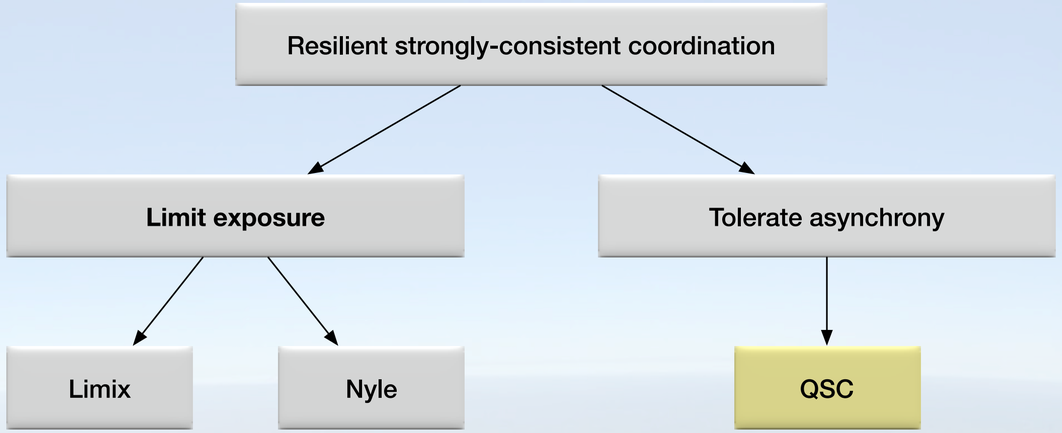


QSC roadmap



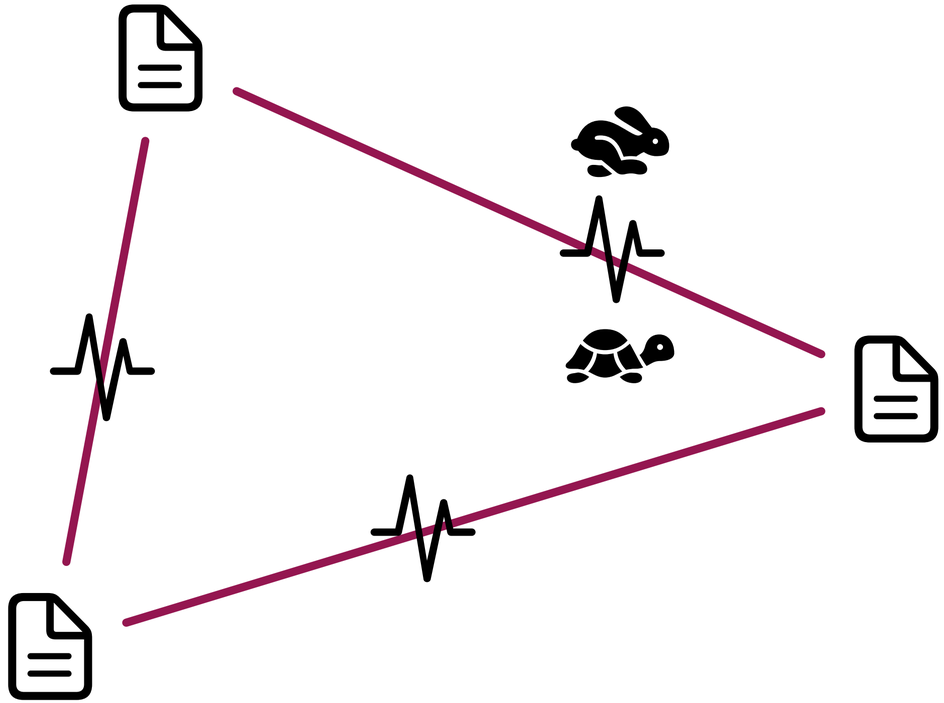
- Motivation and setup
- Asynchronous consensus designs
- QSC insights
- Threshold-synchronous broadcast
- The QSC algorithm
- Complexity and optimizations
- Summary

QSC roadmap

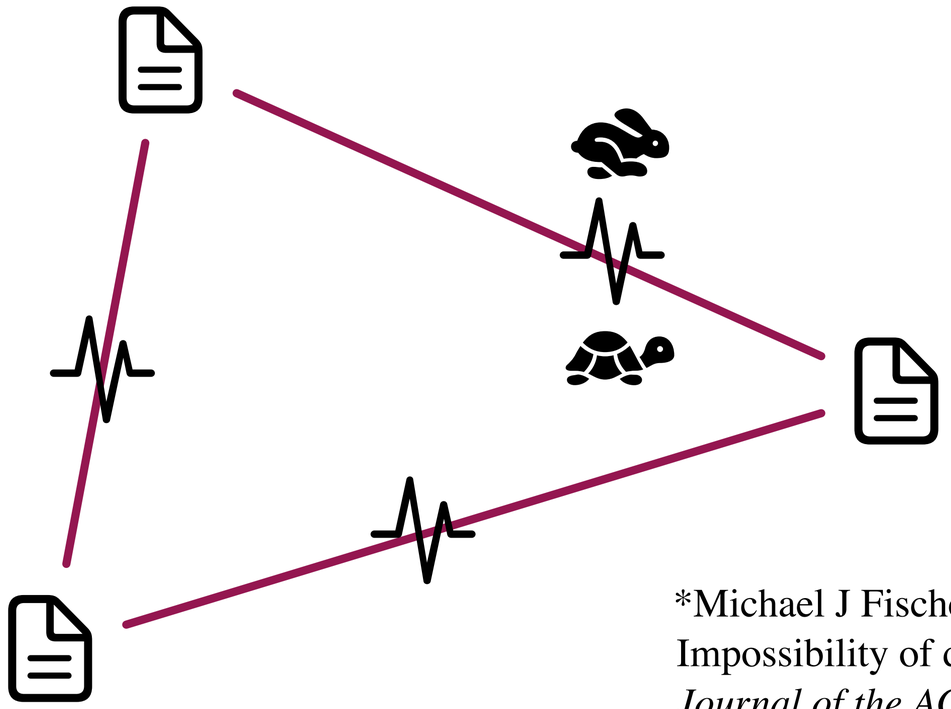


- **Motivation and setup**
- Asynchronous consensus designs
- QSC insights
- Threshold-synchronous broadcast
- The QSC algorithm
- Complexity and optimizations
- Summary

Coordination under asynchrony



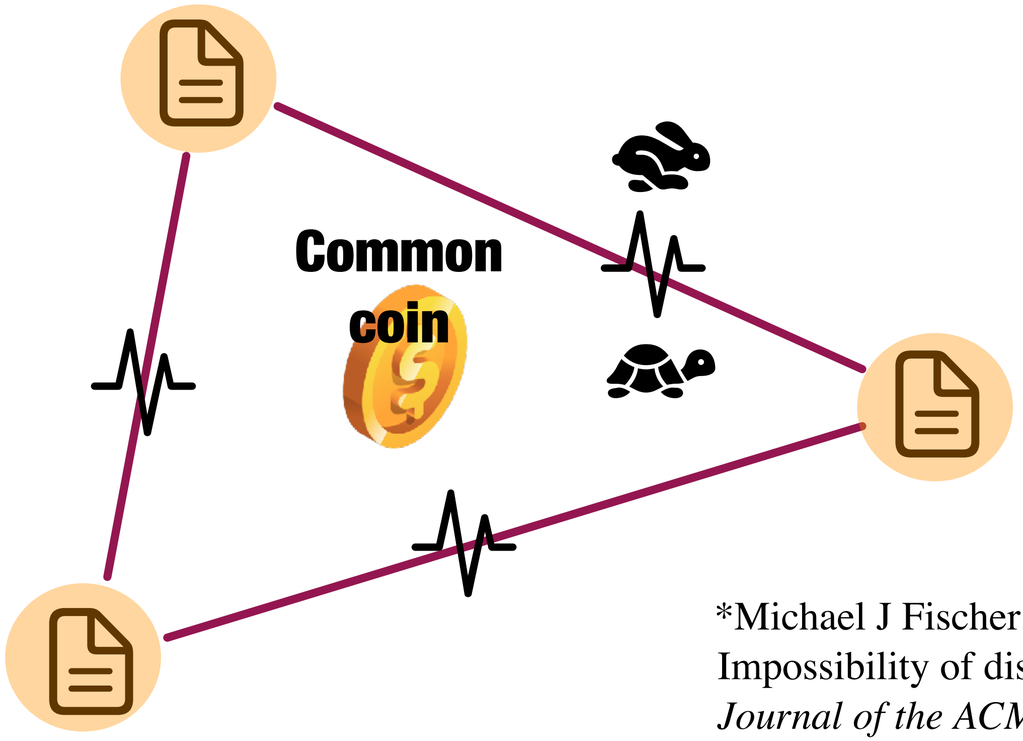
Coordination under asynchrony



*Michael J Fischer, Nancy A Lynch, and Michael S Paterson. Impossibility of distributed consensus with one faulty process. *Journal of the ACM (JACM)*, 32(2):374–382, 1985.

Deterministic consensus cannot guarantee progress (FLP impossibility result*)

Coordination under asynchrony

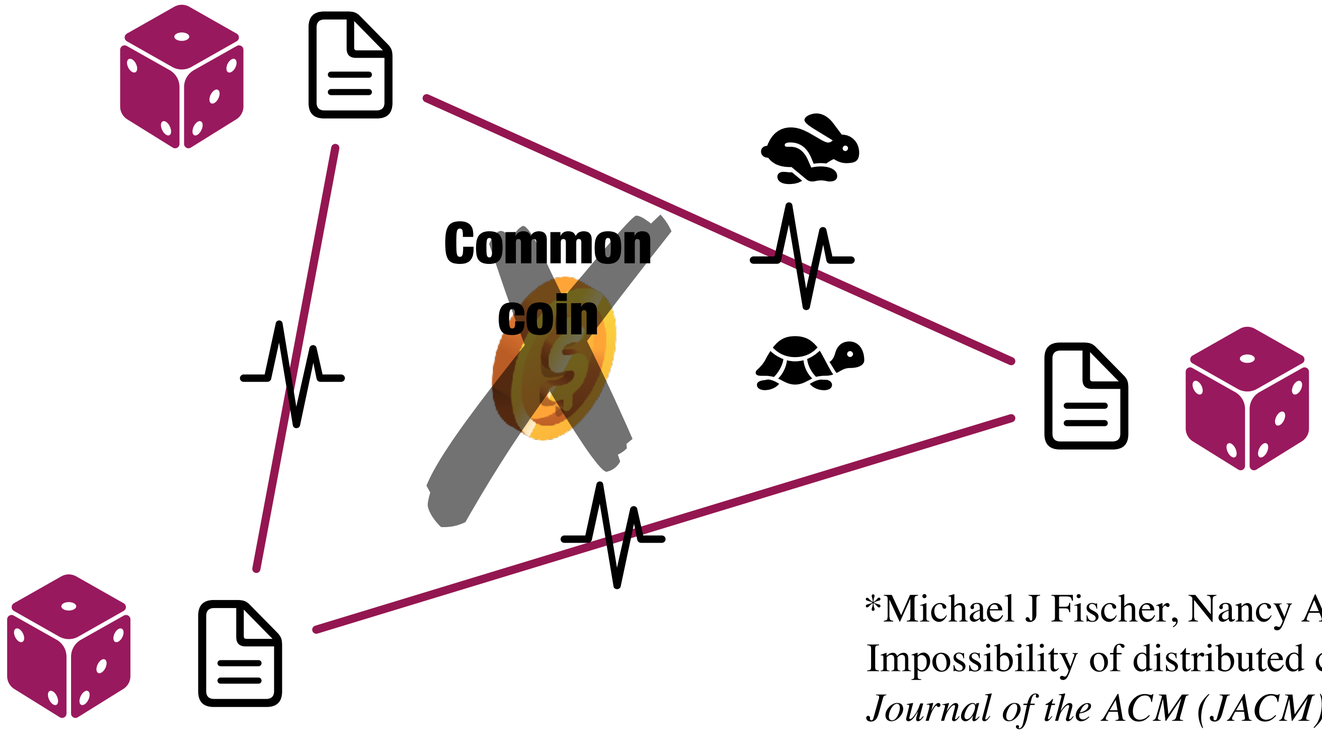


*Michael J Fischer, Nancy A Lynch, and Michael S Paterson. Impossibility of distributed consensus with one faulty process. *Journal of the ACM (JACM)*, 32(2):374–382, 1985.

Deterministic consensus cannot guarantee progress (FLP impossibility result*)

Randomized consensus can, but is rarely deployed

Coordination under asynchrony



*Michael J Fischer, Nancy A Lynch, and Michael S Paterson. Impossibility of distributed consensus with one faulty process. *Journal of the ACM (JACM)*, 32(2):374–382, 1985.

Deterministic consensus cannot guarantee progress (FLP impossibility result*)

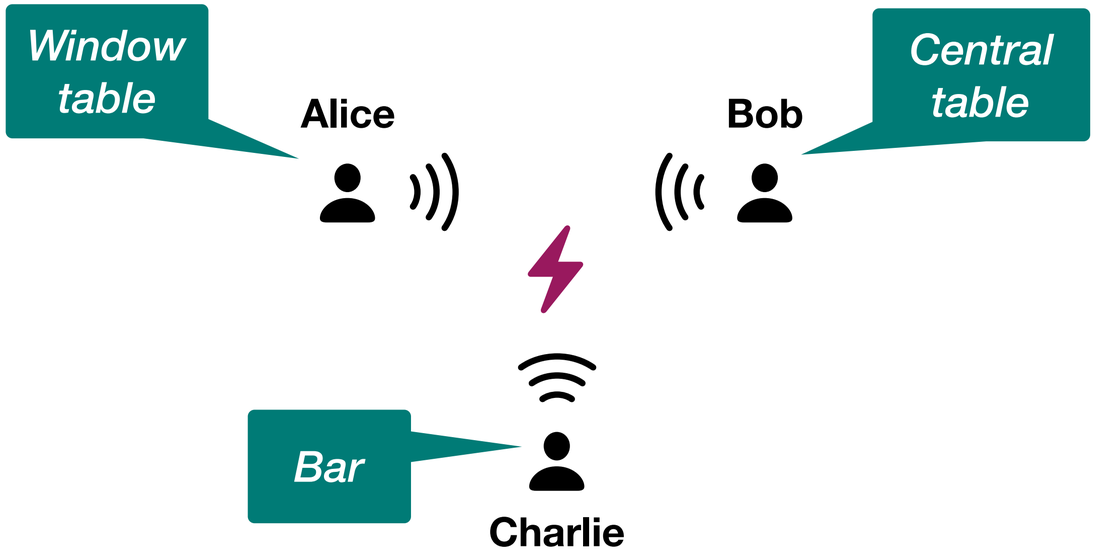
Randomized consensus can, but is rarely deployed

QSC: use private randomness (simpler) instead of a common coin

Consensus setup

- *“What table shall we sit at?”*
- **N** nodes, a minority **f** can fail by **crashing (fail-silent)**
- Each node proposes a value (multi-value consensus)
- Messages eventually delivered, but after a finite arbitrary delay

Goals



Consensus setup

- *“What table shall we sit at?”*
- **N** nodes, a minority **f** can fail by **crashing (fail-silent)**
- Each node proposes a value (multi-value consensus)
- Messages eventually delivered, but after a finite arbitrary delay

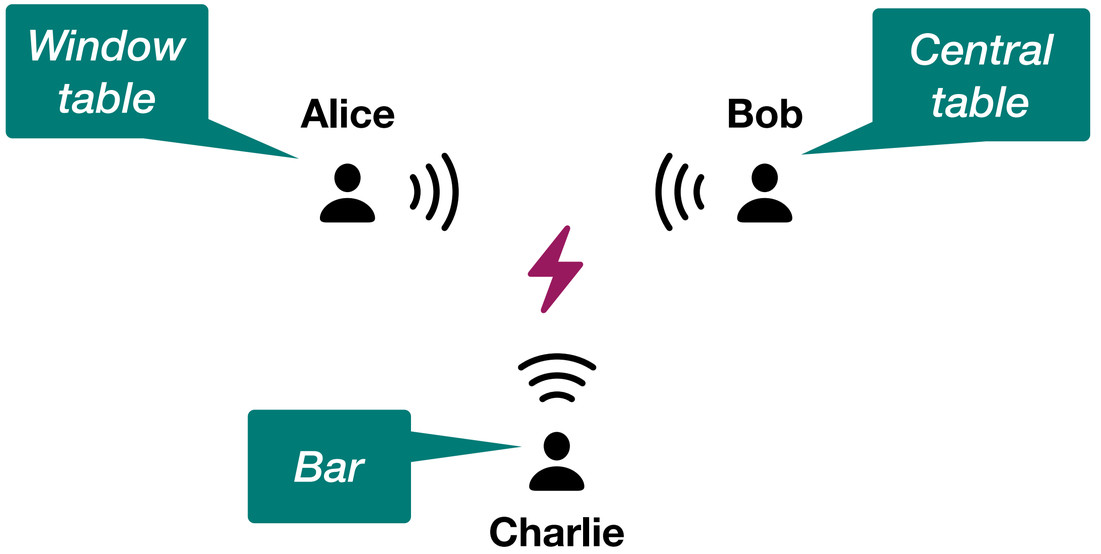
Goals

Termination: Every live node eventually decides

Agreement: No two nodes decide differently

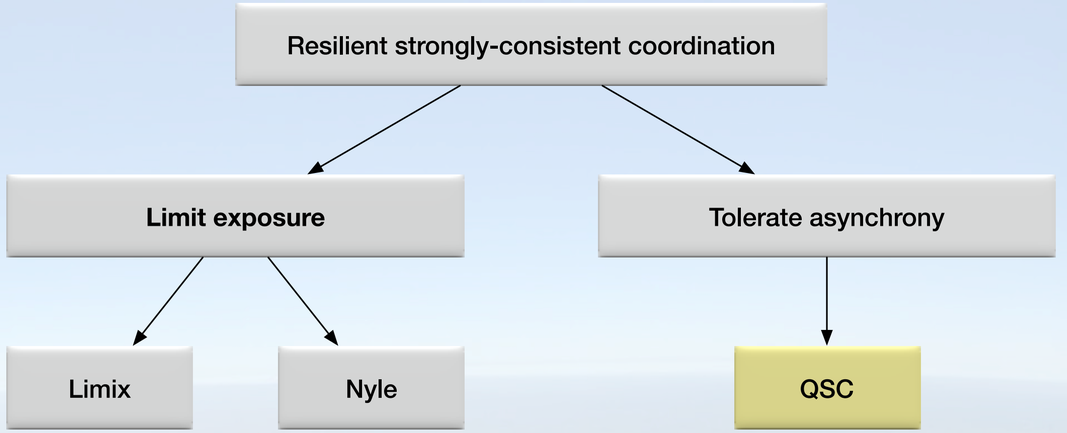
Validity: The value decided is a proposed value

Integrity: No node decides twice

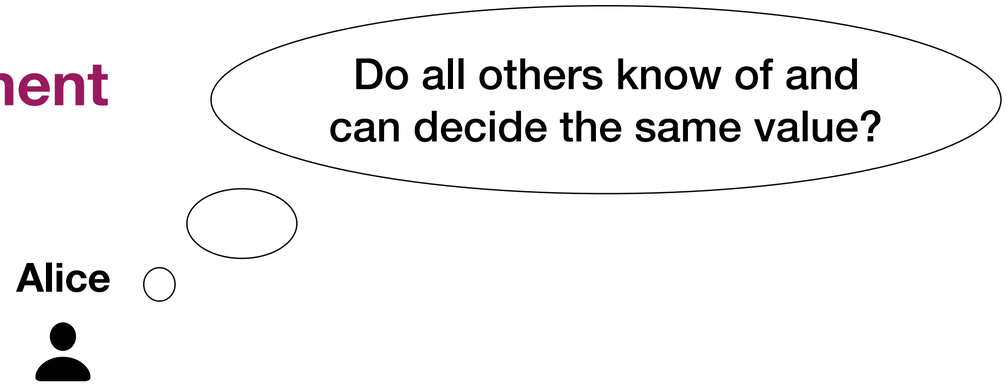


QSC roadmap

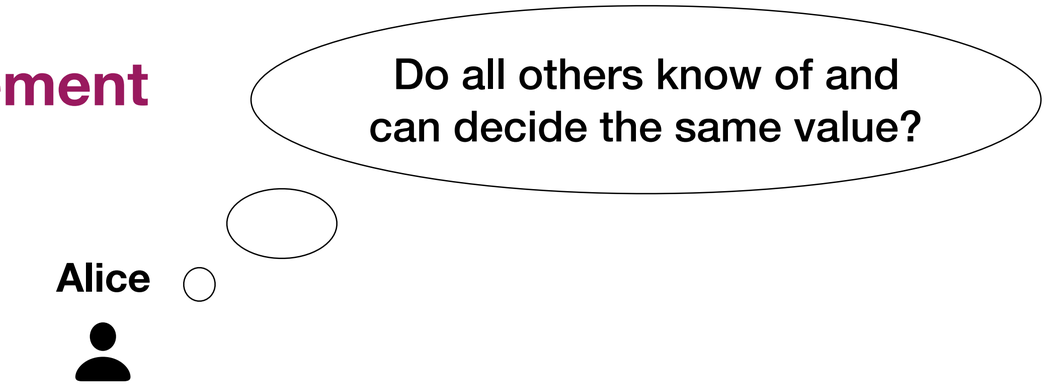
- Motivation and setup
- **Asynchronous consensus designs**
  - **General steps**
  - Binary consensus using common core (Attiya-Welch)
  - Consensus using reliable broadcast
- QSC insights
- Threshold-synchronous broadcast
- The QSC algorithm
- Complexity and optimizations
- Summary



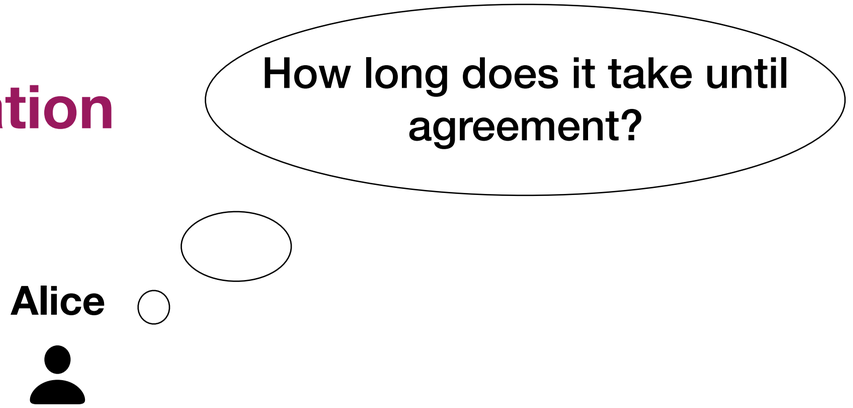
Agreement



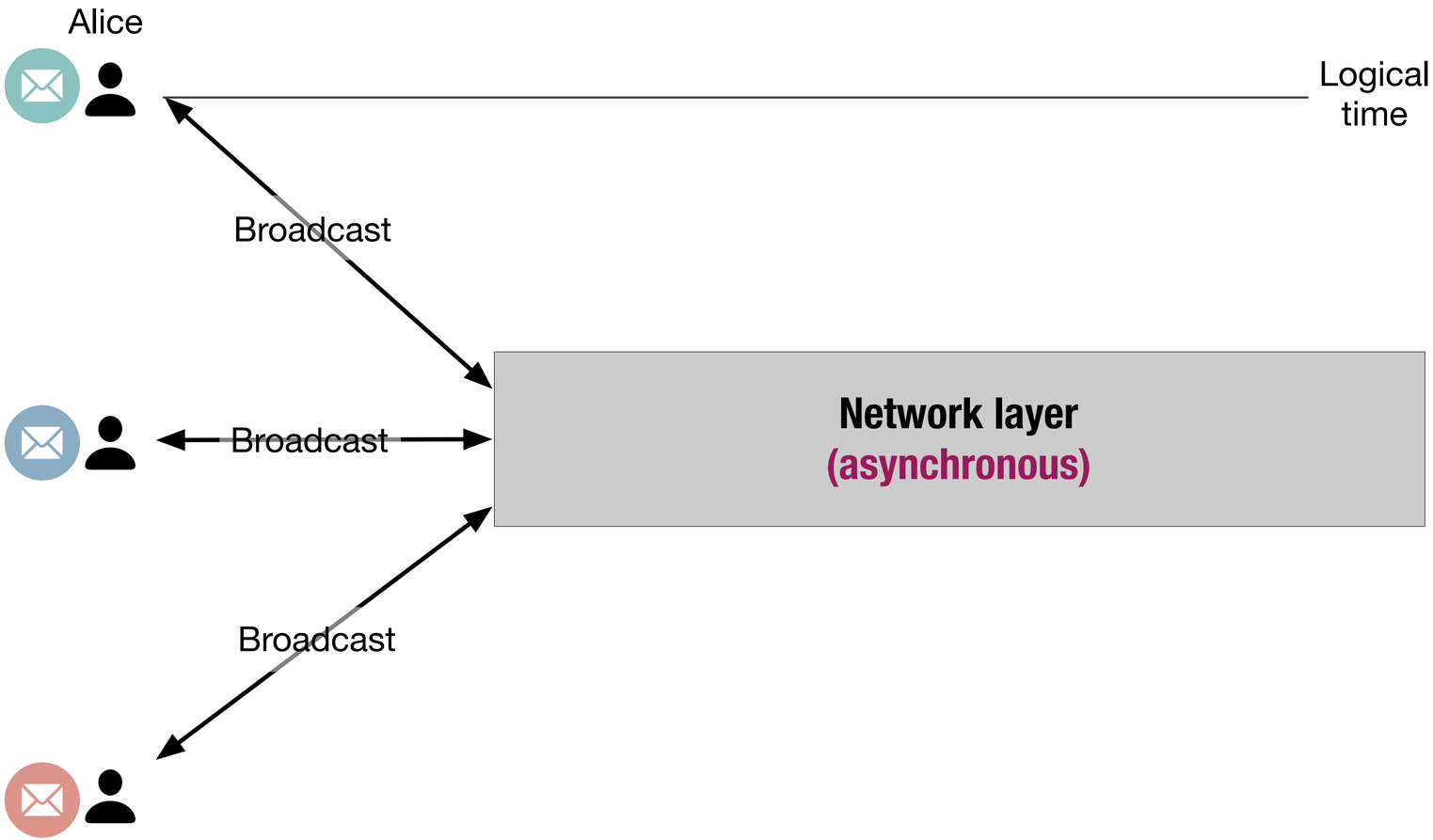
Agreement



Termination



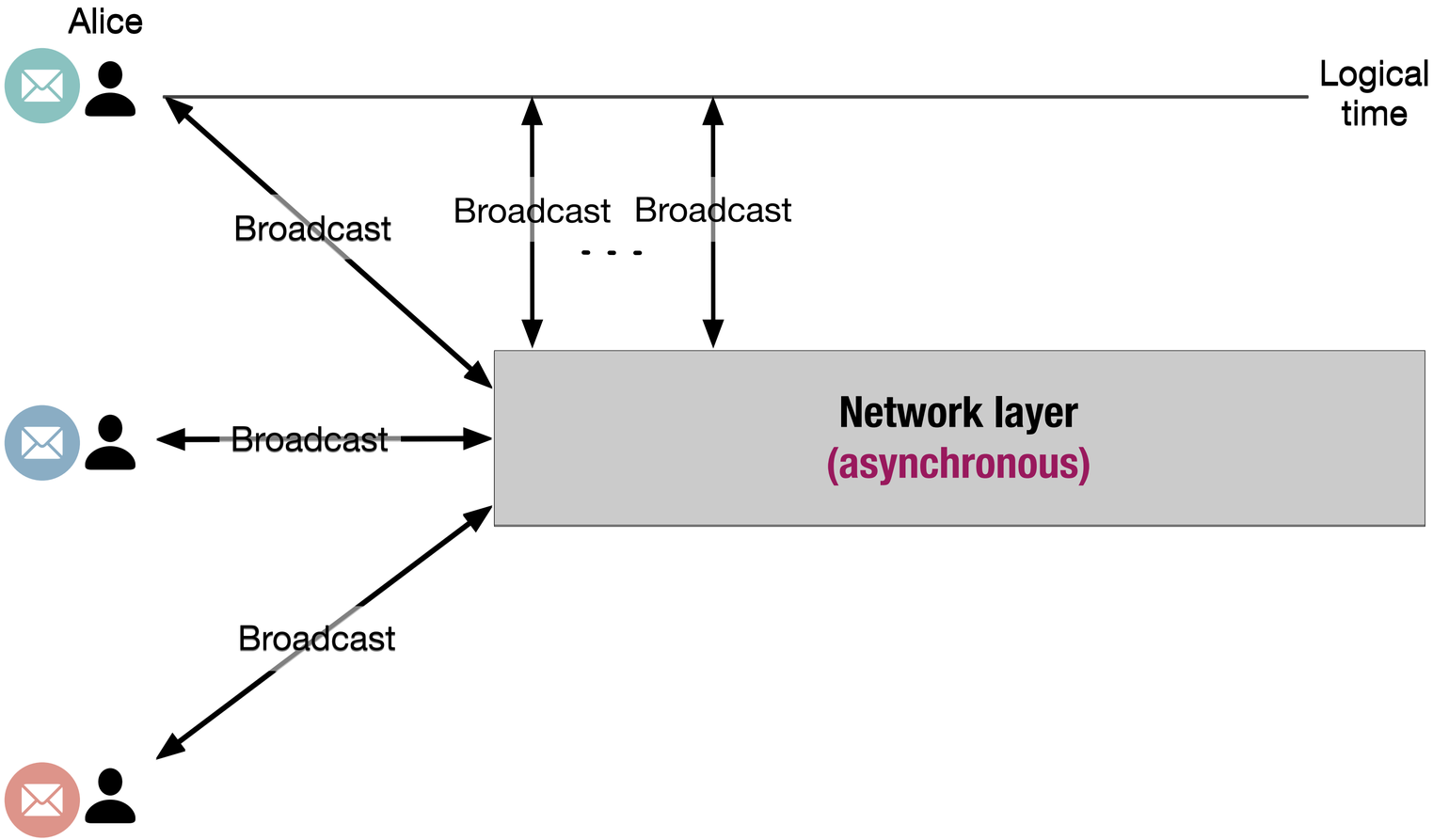
General steps



Exchange of information

Needed for validity

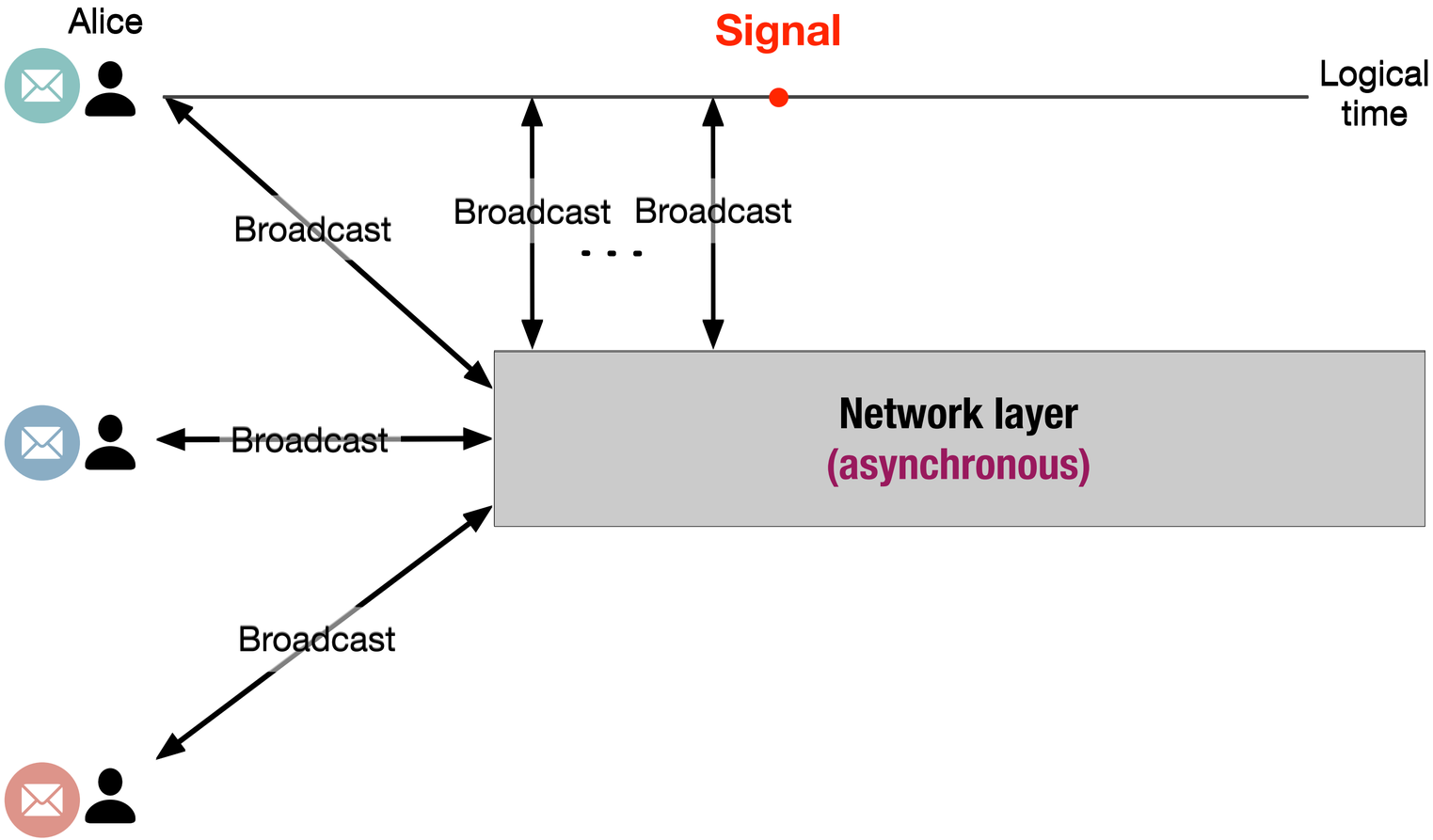
General steps



Exchange of information

Needed for validity

General steps



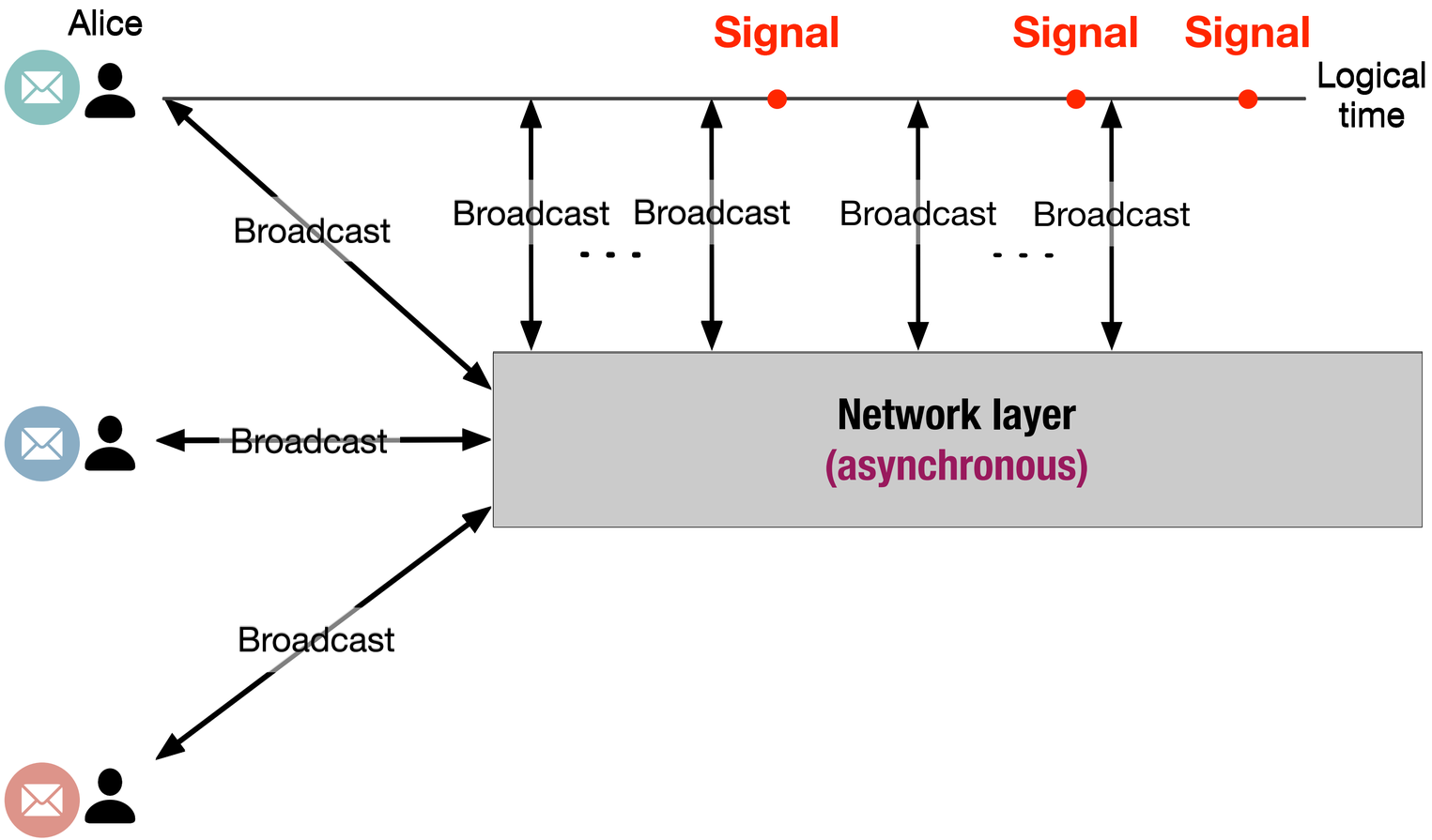
Exchange of information

Needed for validity

Wait for signal

A condition potentially useful for agreement

General steps



Exchange of information

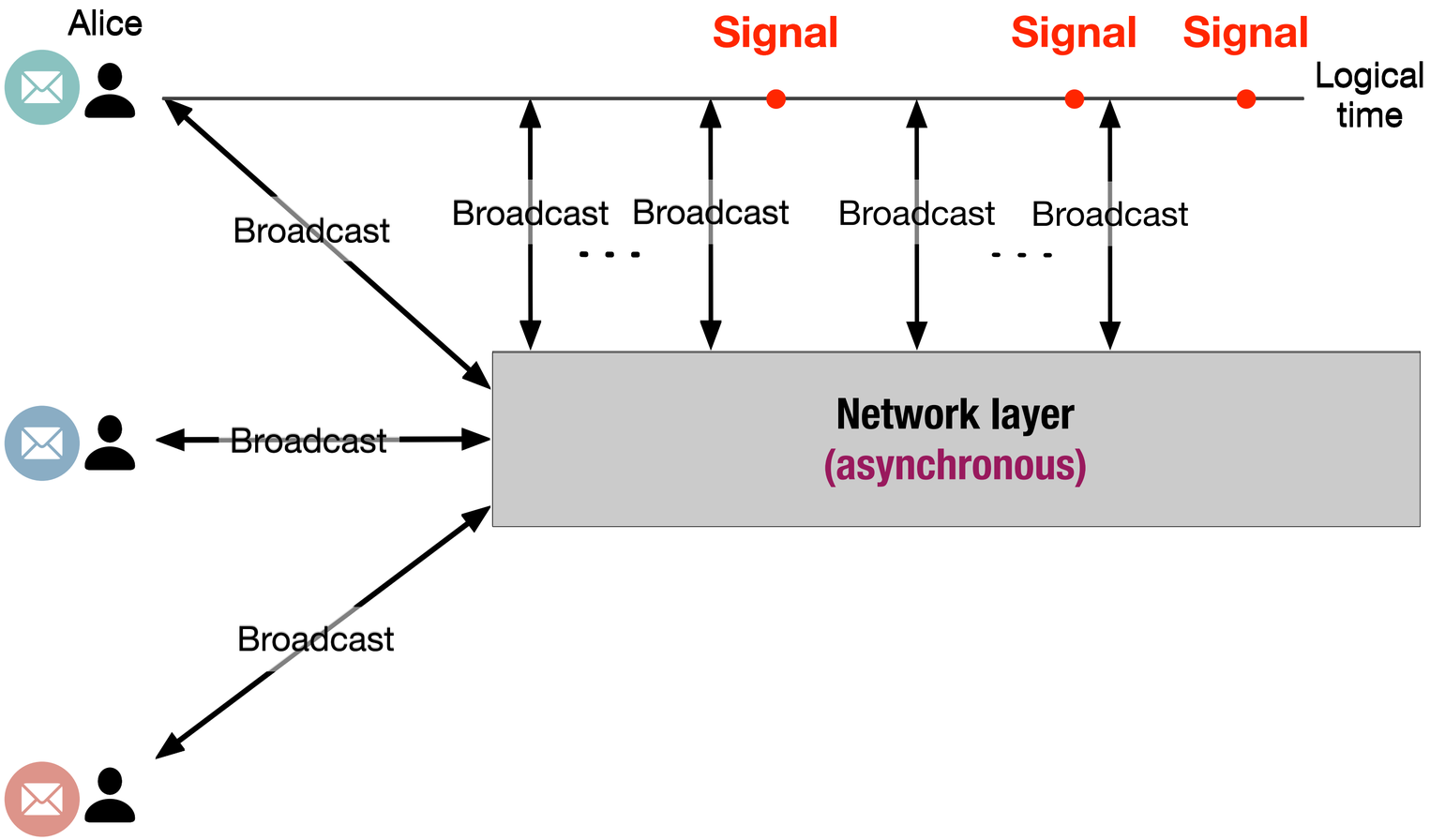
Needed for validity

Wait for signal

A condition potentially useful for agreement

Requires randomness

General steps



Exchange of information

Needed for validity

Wait for signal

A condition potentially useful for agreement

Requires randomness

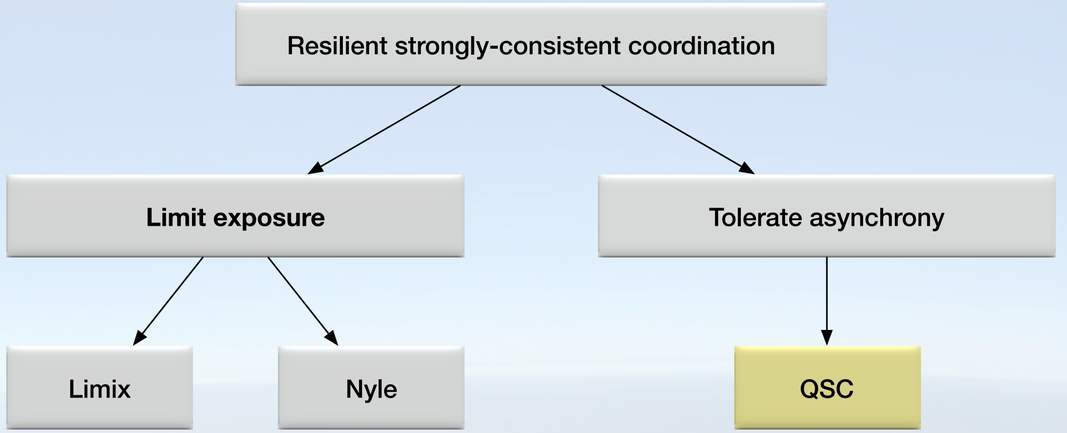
Termination

signal steps necessary to decide

What does the signal involve?

QSC roadmap

- Motivation and setup
- **Asynchronous consensus designs**
  - General steps
  - **Binary consensus using common core (Attiya-Welch)**
  - Consensus using reliable broadcast
- QSC insights
- Threshold-synchronous broadcast
- The QSC algorithm
- Complexity and optimizations
- Summary



Binary consensus using common core*

Agree on 0 or 1

Exchange of information akin to voting

Use common core towards ensuring agreement

*Hagit Attiya, Jennifer Welch. *Distributed Computing: Fundamentals, Simulations and Advanced Topics*. John Wiley & Sons, Inc., Hoboken, NJ, USA, 2004

Binary consensus using common core*

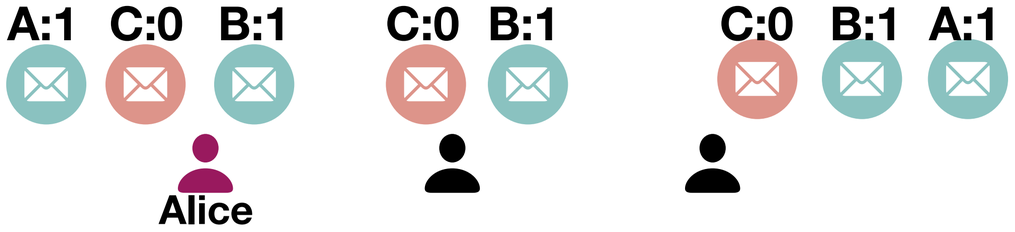
Agree on 0 or 1

Exchange of information akin to voting

Use common core towards ensuring agreement

Common core

N-f proposals seen by all live nodes



*Hagit Attiya, Jennifer Welch. *Distributed Computing: Fundamentals, Simulations and Advanced Topics*. John Wiley & Sons, Inc., Hoboken, NJ, USA, 2004

Binary consensus using common core*

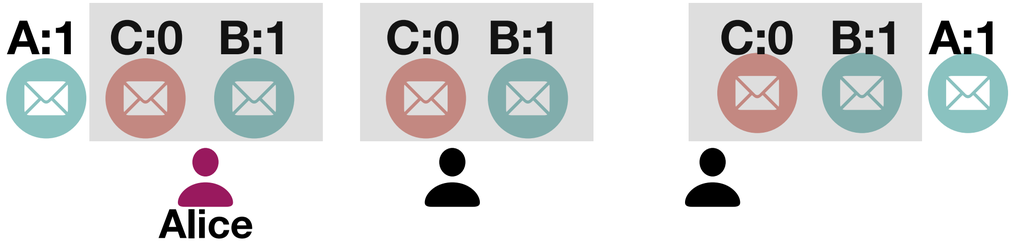
Agree on 0 or 1

Exchange of information akin to voting

Use common core towards ensuring agreement

Common core

N-f proposals seen by all live nodes



*Hagit Attiya, Jennifer Welch. *Distributed Computing: Fundamentals, Simulations and Advanced Topics*. John Wiley & Sons, Inc., Hoboken, NJ, USA, 2004

Binary consensus using common core*

Agree on 0 or 1

Exchange of information akin to voting

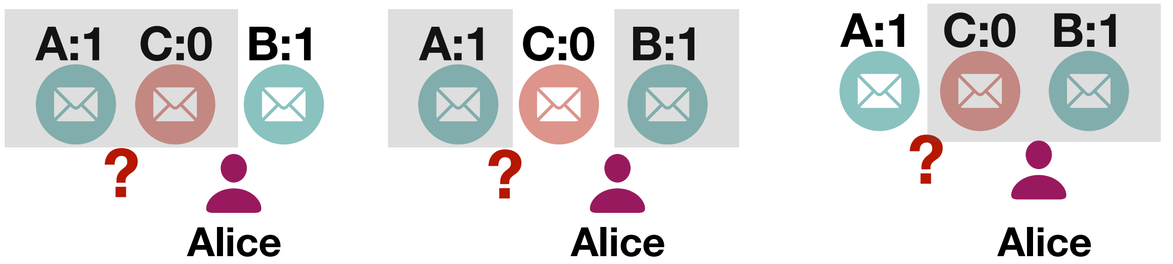
Use common core towards ensuring agreement

Common core

N-f proposals seen by all live nodes

Cannot identify proposals in the common core

Relies on supermajorities to decide



*Hagit Attiya, Jennifer Welch. *Distributed Computing: Fundamentals, Simulations and Advanced Topics*. John Wiley & Sons, Inc., Hoboken, NJ, USA, 2004

Binary consensus using common core*

Agree on 0 or 1

Exchange of information akin to voting

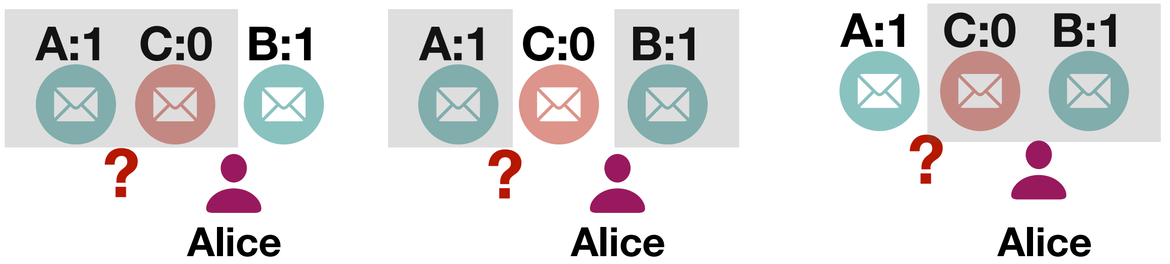
Use common core towards ensuring agreement

Common core

N-f proposals seen by all live nodes

Cannot identify proposals in the common core

Relies on supermajorities to decide



Signal

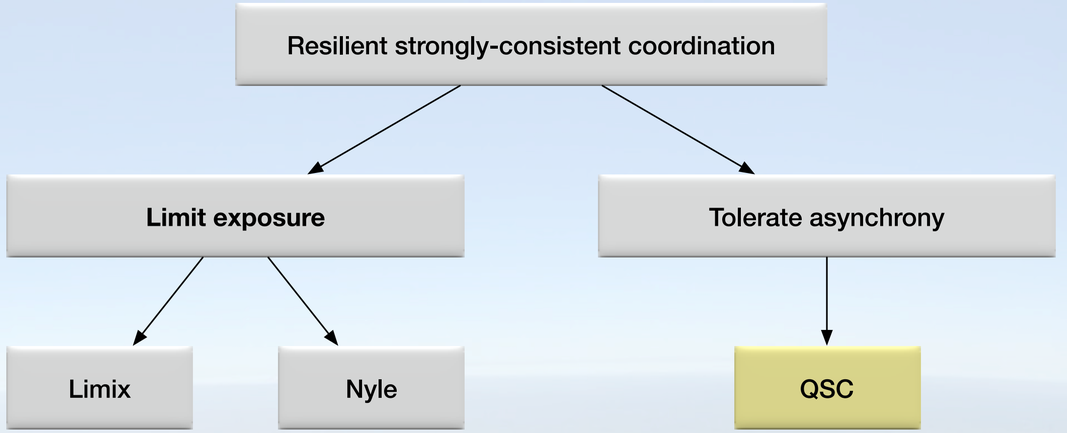
Common core and supermajorities

Falls back on **common coin**

*Hagit Attiya, Jennifer Welch. *Distributed Computing: Fundamentals, Simulations and Advanced Topics*. John Wiley & Sons, Inc., Hoboken, NJ, USA, 2004

QSC roadmap

- Motivation and setup
- **Asynchronous consensus designs**
  - General steps
  - Binary consensus using common core (Attiya-Welch)
  - **Consensus using reliable broadcast**
- QSC insights
- Threshold-synchronous broadcast
- The QSC algorithm
- Complexity and optimizations
- Summary



Consensus using reliable broadcast

Agreement on arbitrary values

Cannot use majority vote

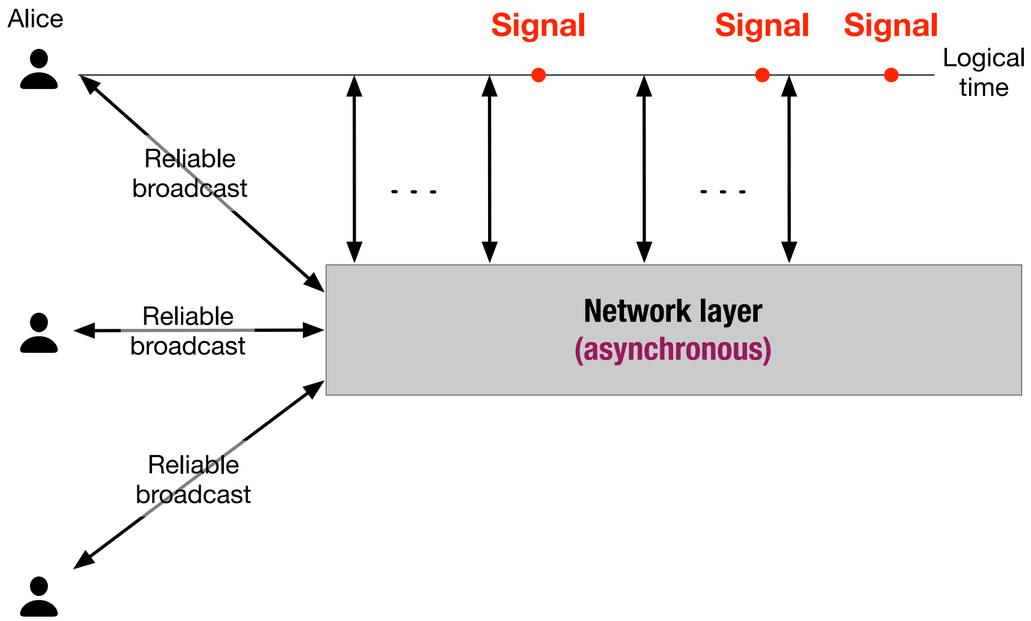
Consensus using reliable broadcast

Agreement on arbitrary values

Cannot use majority vote

Use reliable broadcast

Eventually all live nodes receive the same proposals.
But when?



Consensus using reliable broadcast

Agreement on arbitrary values

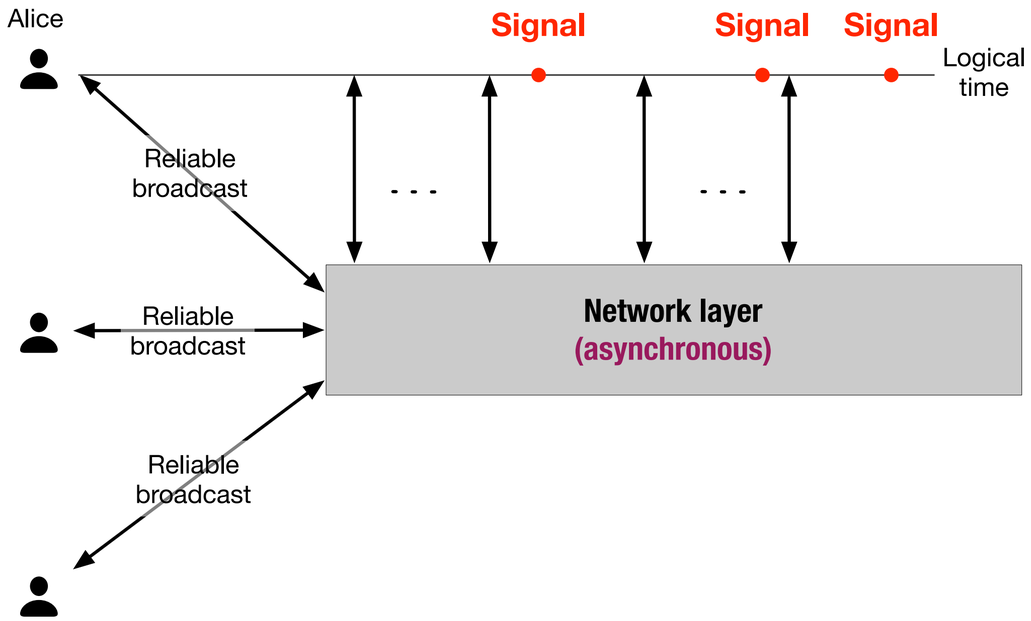
Cannot use majority vote

Use reliable broadcast

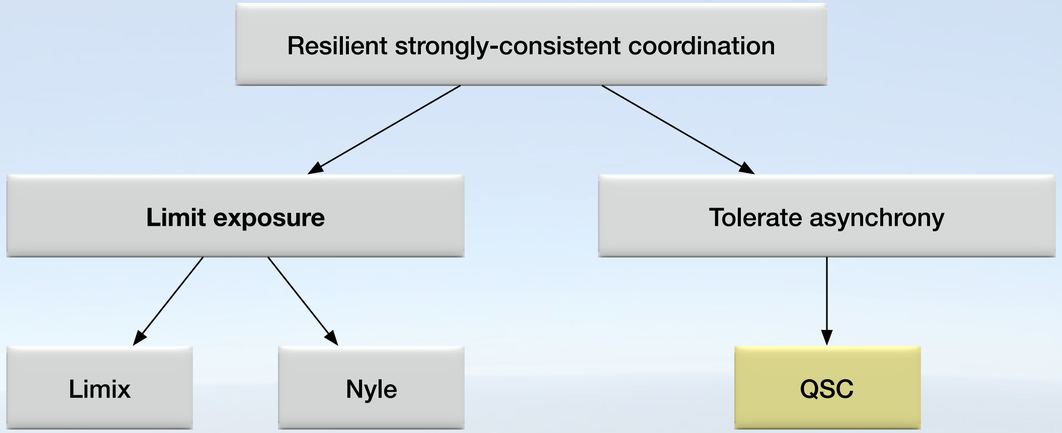
Eventually all live nodes receive the same proposals.
But when?

A possible signal

N binary consensus blocks



QSC roadmap



- Motivation and setup
- Asynchronous consensus designs
- **QSC insights**
- Threshold-synchronous broadcast
- The QSC algorithm
- Complexity and optimizations
- Summary

QSC insights

Agreement on arbitrary values

QSC insights

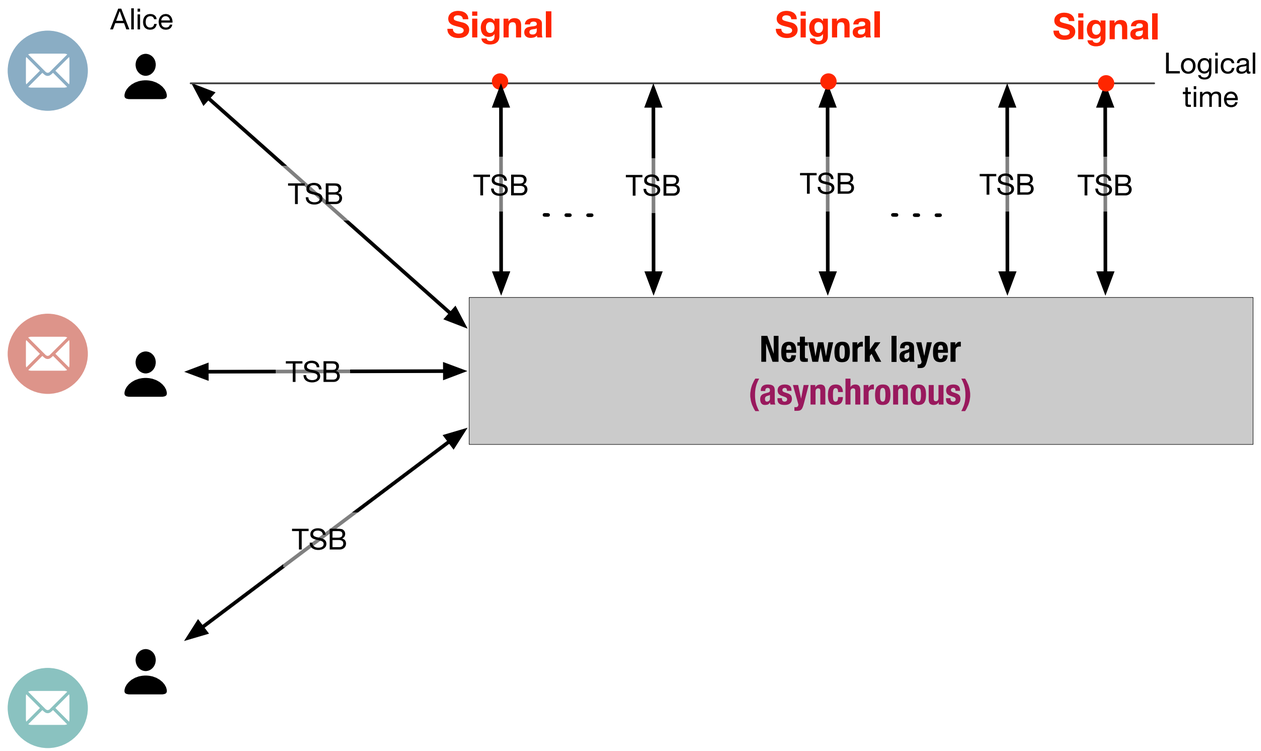
Agreement on arbitrary values

Without needing common coins
or binary consensus

QSC insights

Agreement on arbitrary values

Without needing common coins
or binary consensus

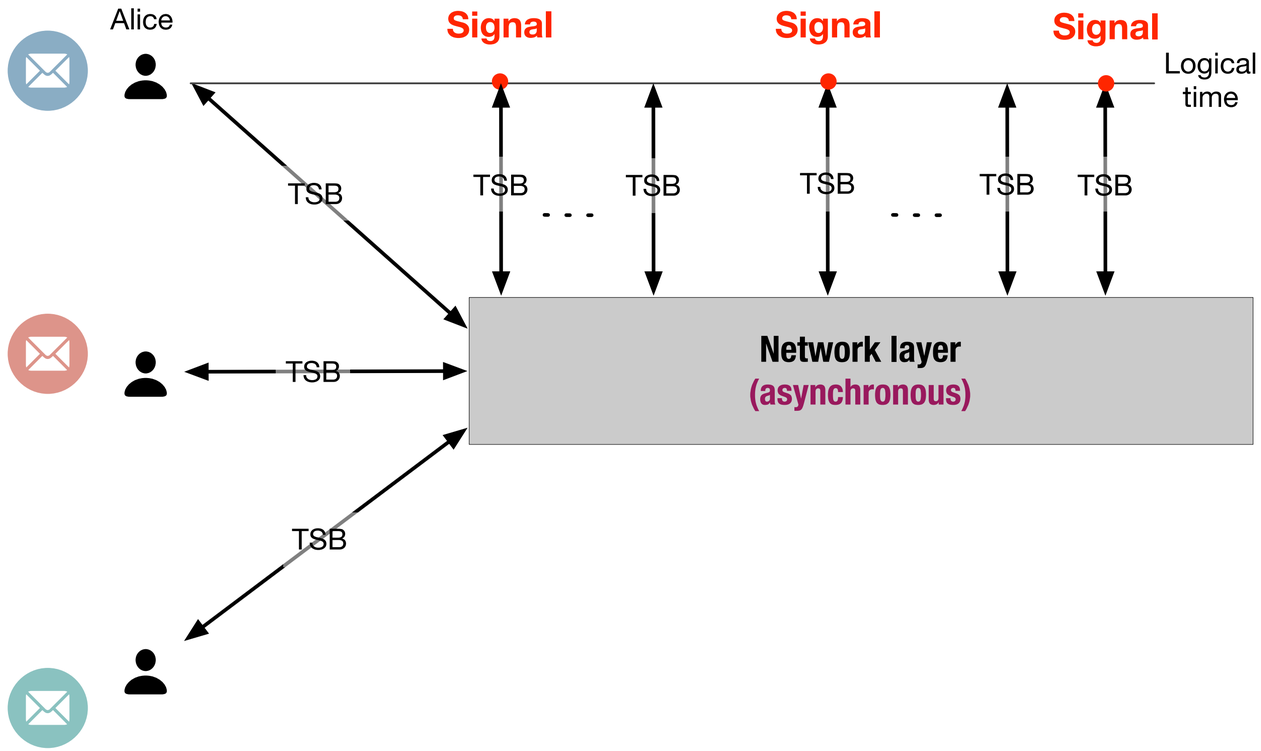


New primitive TSB
(threshold-synchronous broadcast)

QSC insights

Agreement on arbitrary values

Without needing common coins
or binary consensus



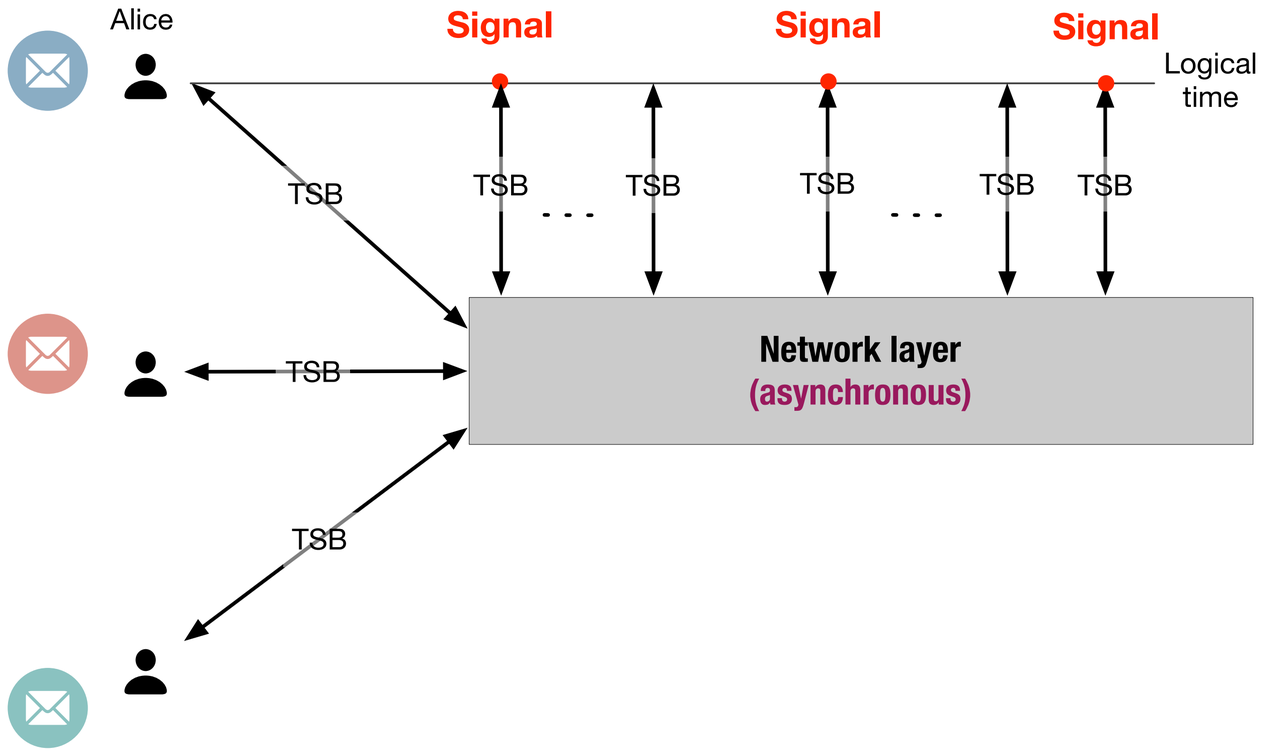
New primitive TSB (threshold-synchronous broadcast)

Unlike reliable broadcast, TSB also provides the **signal**

QSC insights

Agreement on arbitrary values

Without needing common coins
or binary consensus



New primitive TSB (threshold-synchronous broadcast)

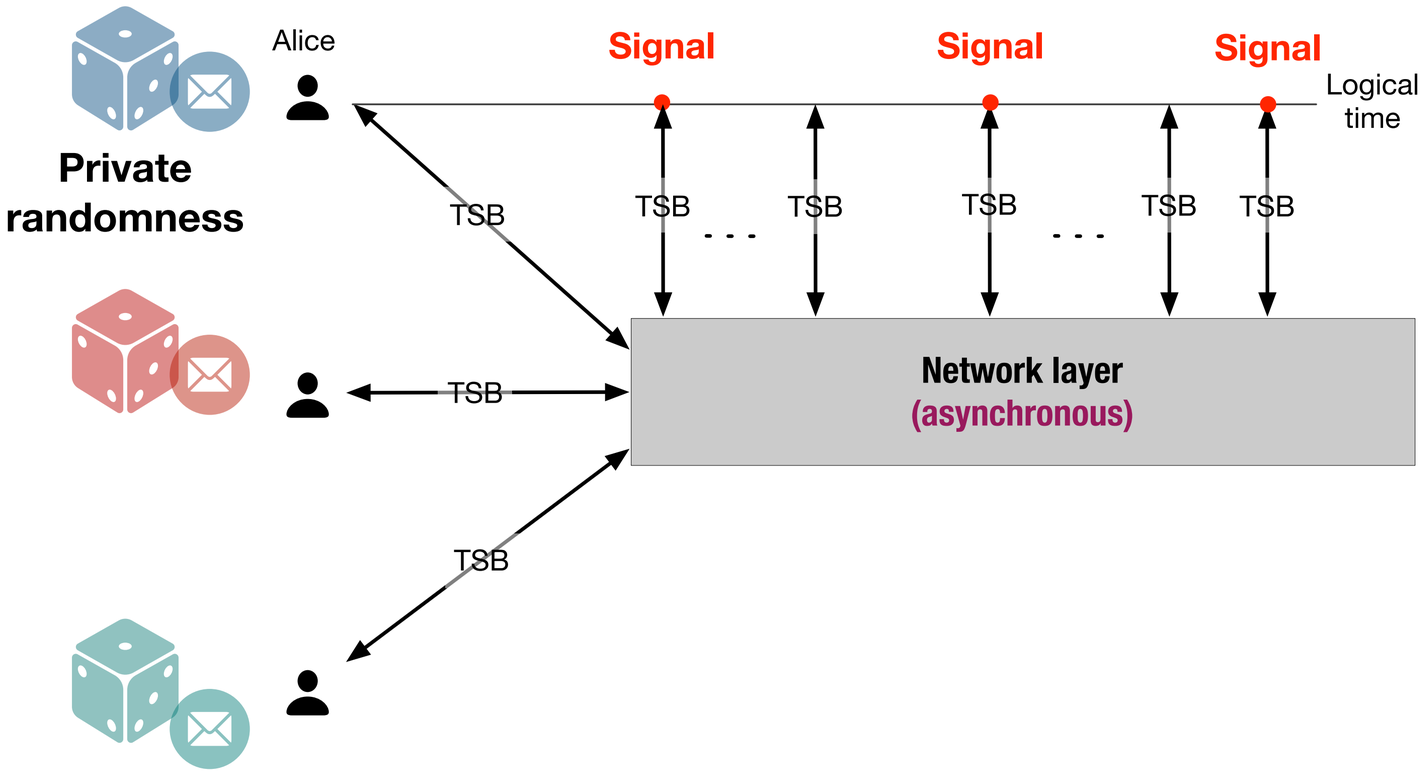
Unlike reliable broadcast, TSB also provides the **signal**

Unlike common core, TSB identifies specific values received by all

QSC insights

Agreement on arbitrary values

Without needing common coins
or binary consensus

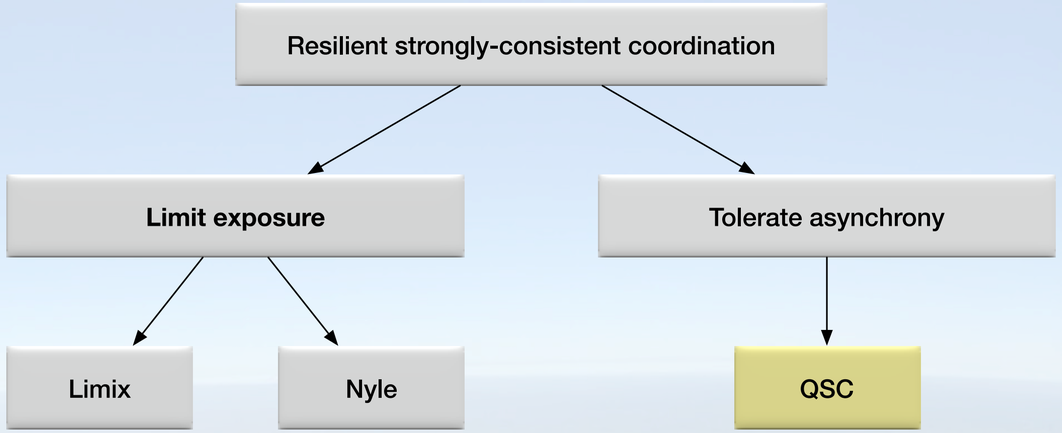


New primitive TSB (threshold-synchronous broadcast)

Unlike reliable broadcast, TSB also provides the **signal**

Unlike common core, TSB identifies specific values received by all

QSC roadmap



- Motivation and setup
- Asynchronous consensus designs
- QSC insights
- Threshold-synchronous broadcast
- The QSC algorithm
- Complexity and optimizations
- **Summary**

QSC summary

- **Crash-fault asynchronous consensus without common coins**
  - Uses node-private randomness
- **TSB: step-synchronous message delivery atop an asynchronous network**

Thesis roadmap

Resilient strongly-consistent coordination

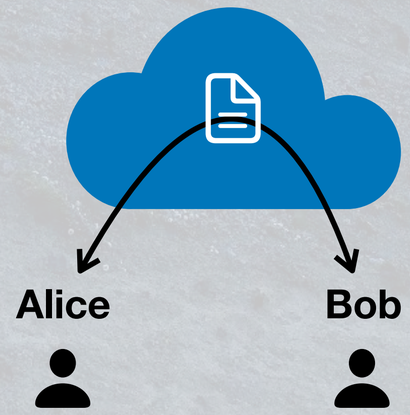
Limit exposure

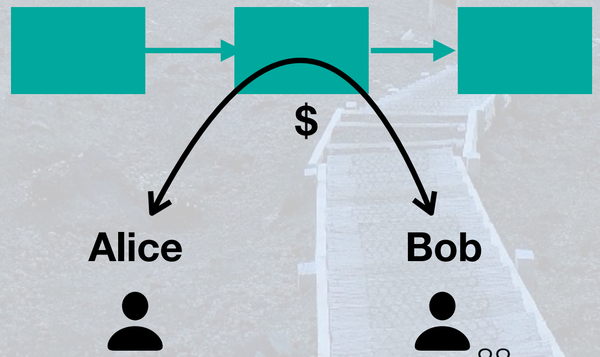
Tolerate asynchrony

Limix

Nyle

QSC





Conclusion

- **Resilience in strongly consistent coordination**
- **Limiting exposure results in systems resilient to distant failures**
  - Limix reduces exposure in metadata services for cloud systems
  - Nyle reduce exposure in BFT-style blockchains through a trust-but-verify approach
- **Tolerating network asynchrony could become more practical**
  - QSC solves asynchronous consensus for crash faults without requiring common coins

Backup slides

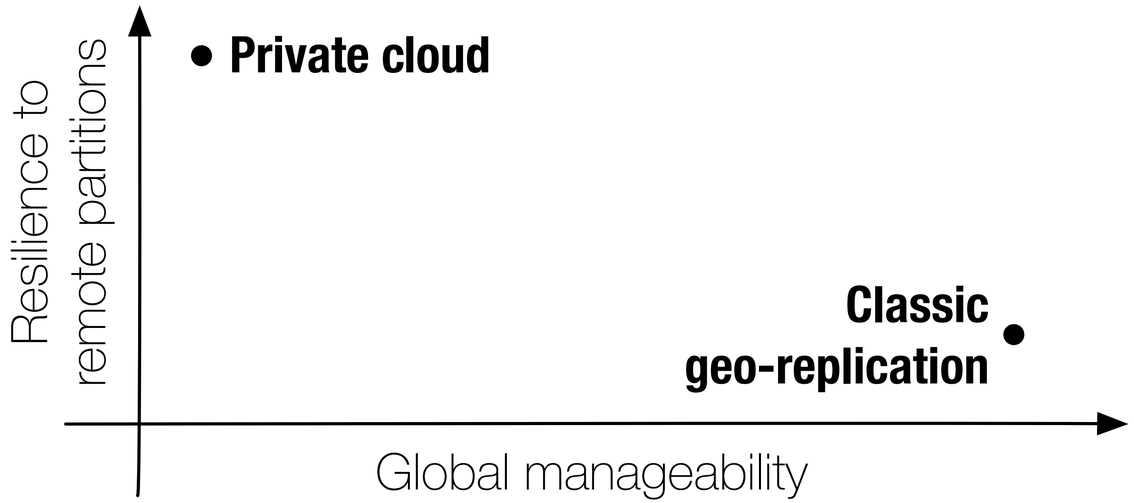
Limix: limit exposure to remote partitions

Limix: limit exposure to remote partitions

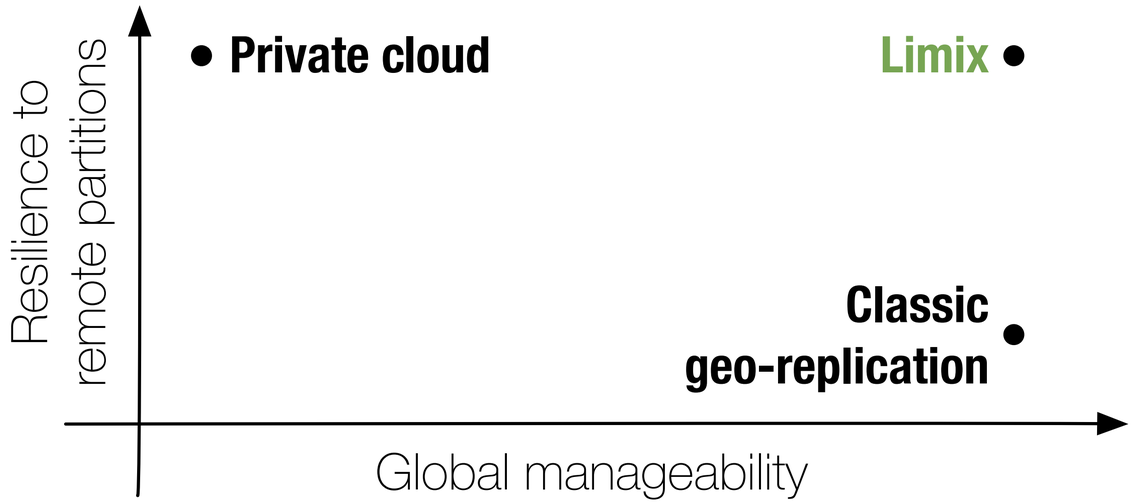
Limix: limit exposure to remote partitions

Limix: limit exposure to remote partitions

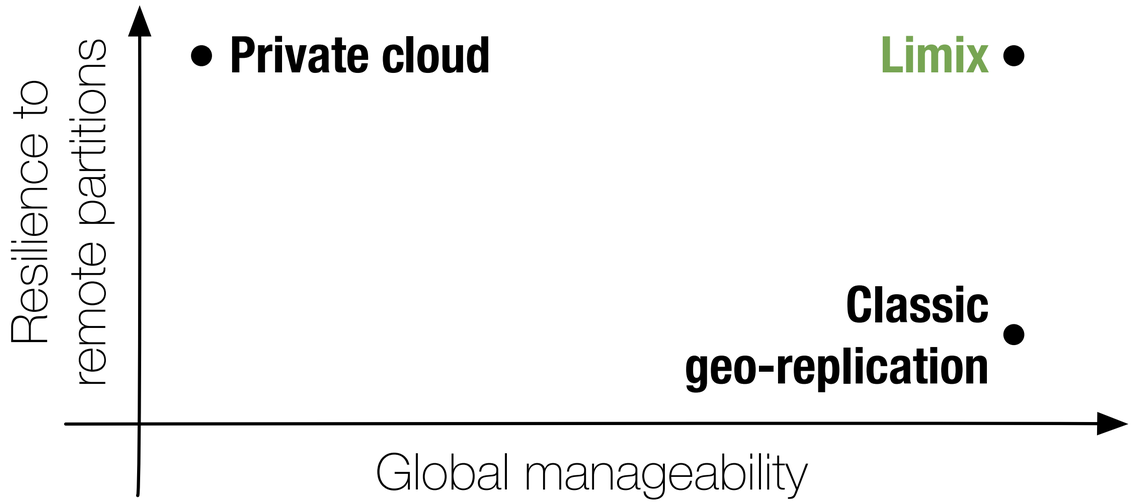
Limix: limit exposure to remote partitions



Limix: limit exposure to remote partitions

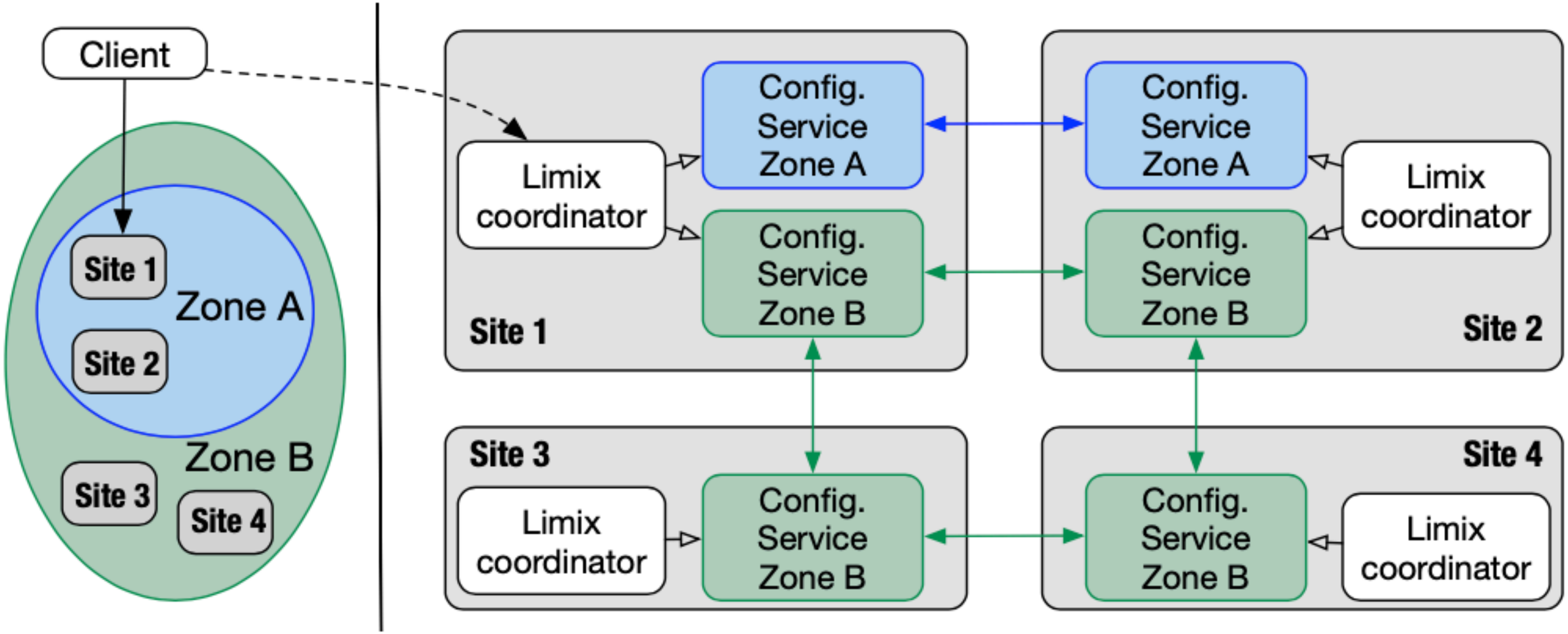


Limix: limit exposure to remote partitions



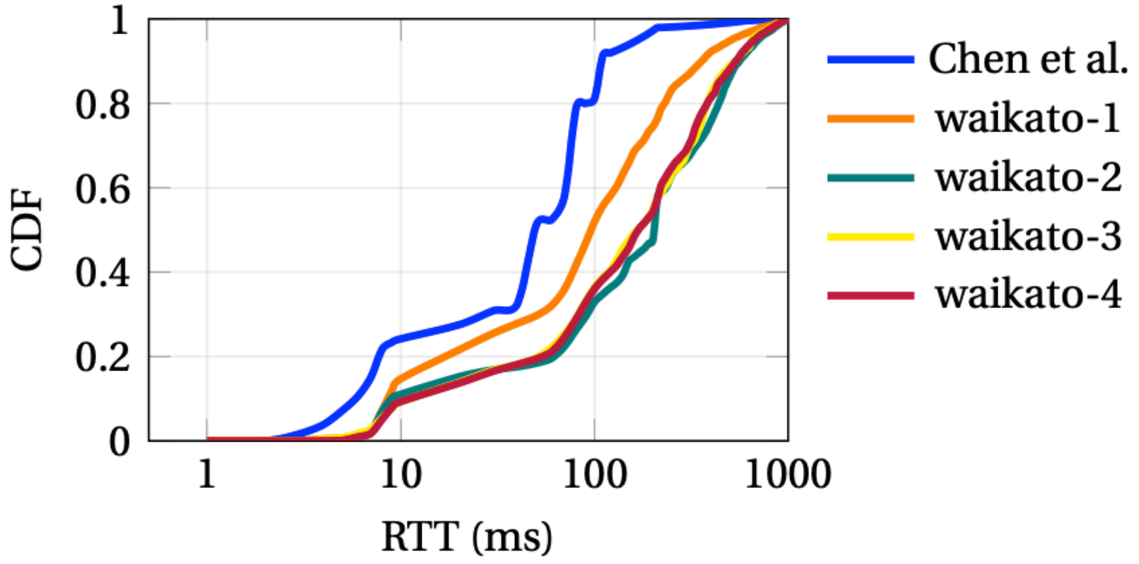
Goal: Limit metadata exposure for all users

Limix architecture

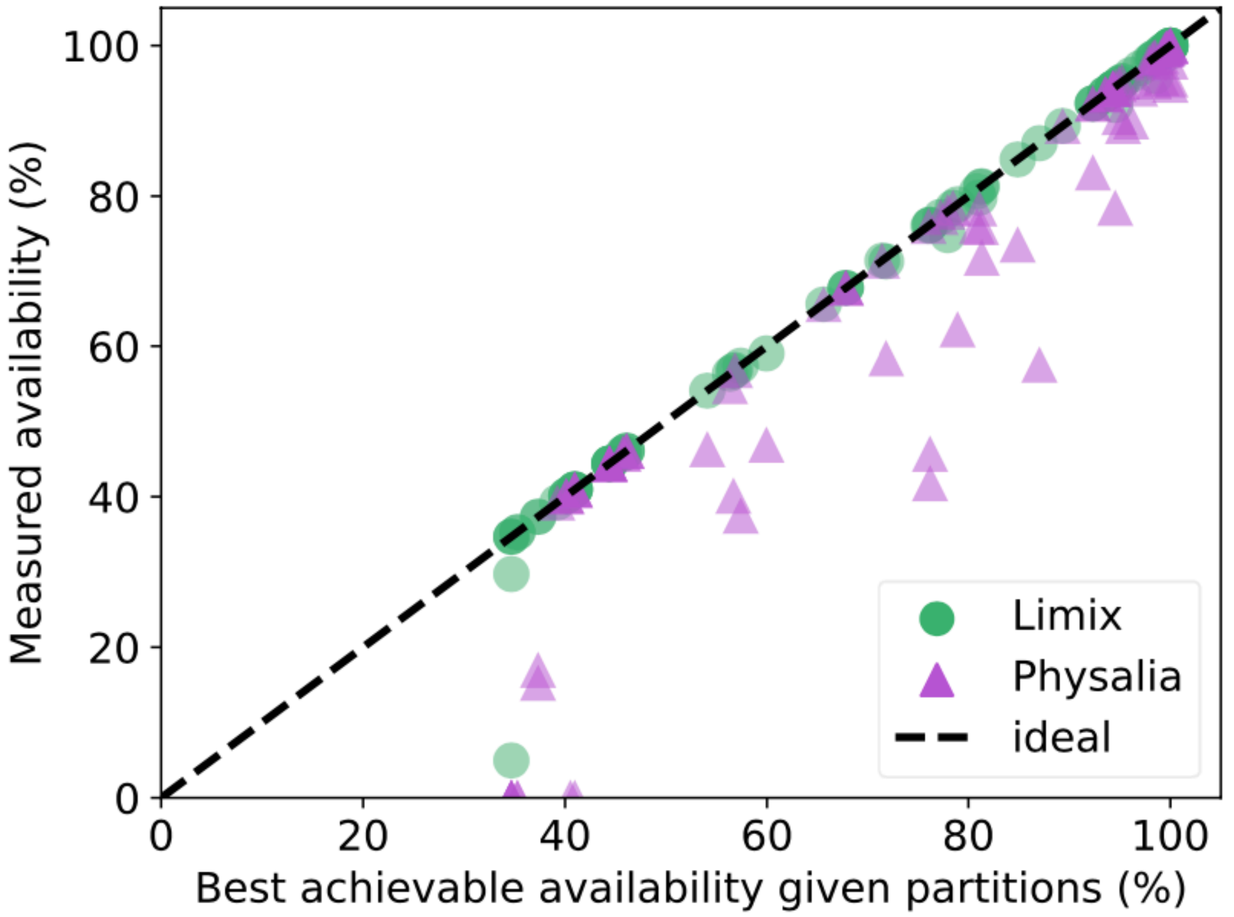


Limix realistic workload experiment

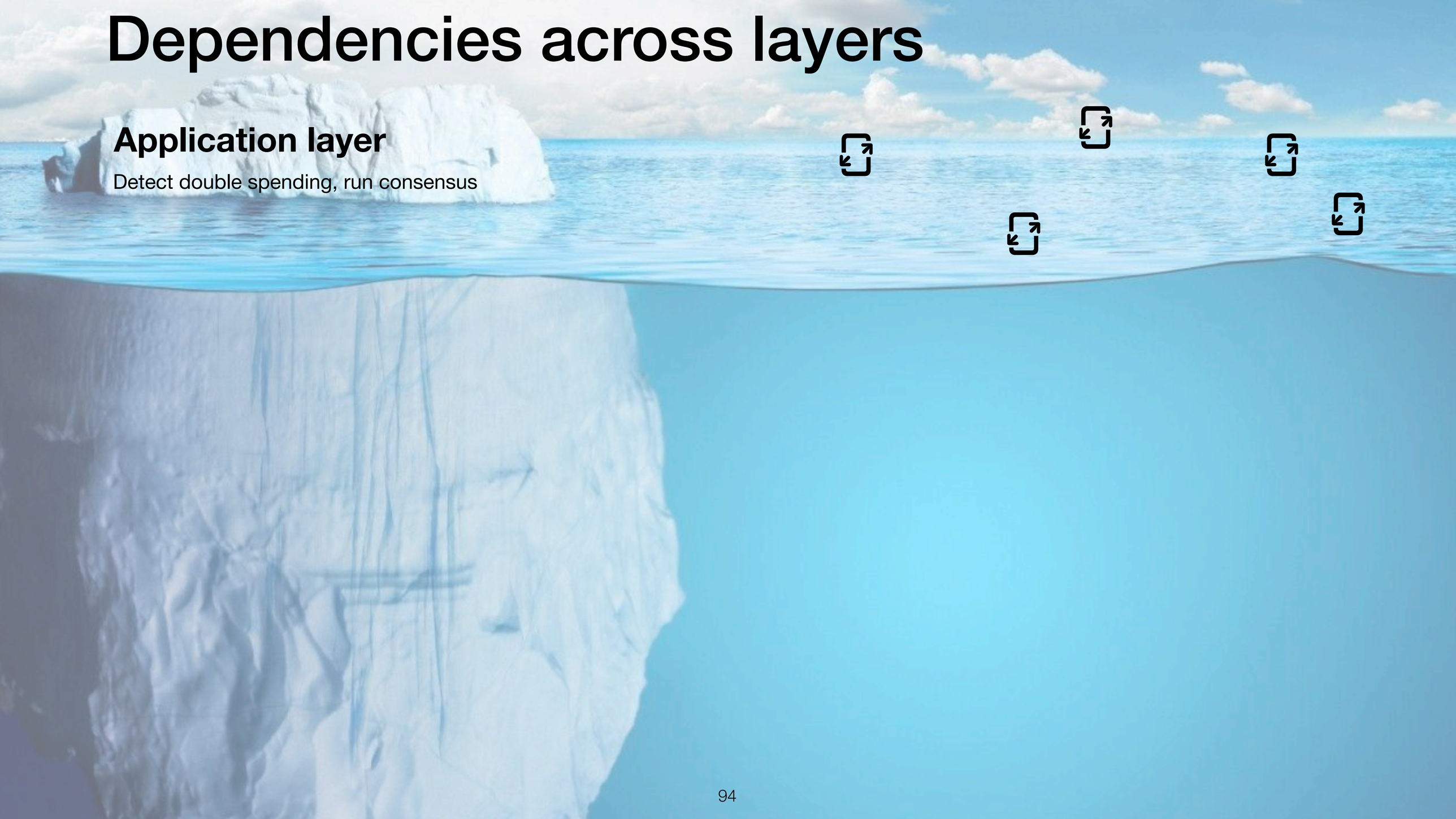
RTT between W-W pairs



Availability



Dependencies across layers



Application layer

Detect double spending, run consensus

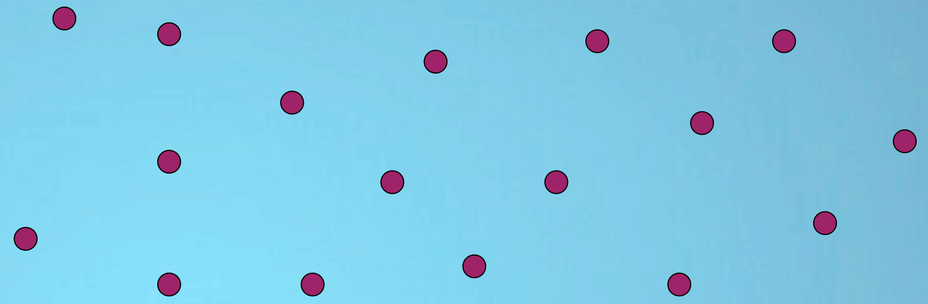
Dependencies across layers

Application layer

Detect double spending, run consensus

Validators layer

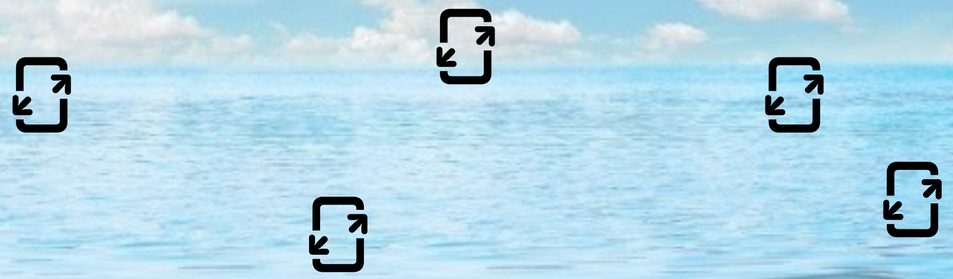
Ensure validator uptime, withstand Byzantine behaviour



Dependencies across layers

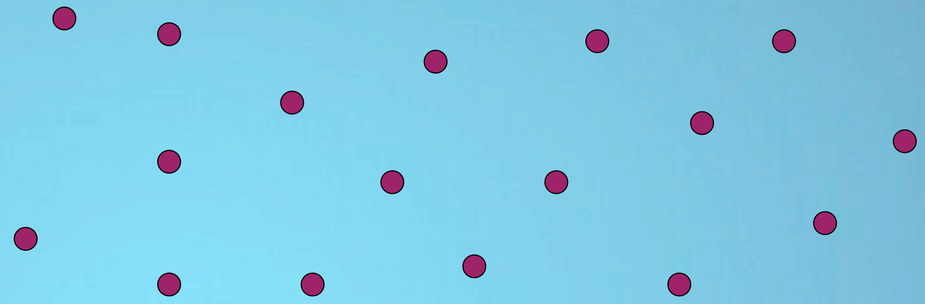
Application layer

Detect double spending, run consensus



Validators layer

Ensure validator uptime, withstand Byzantine behaviour



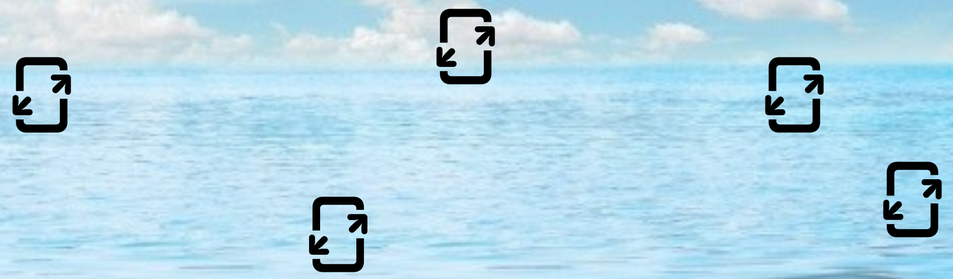
Interconnect layer

Correlated failures and attacks
Are validators up? From whose perspective?

Dependencies across layers

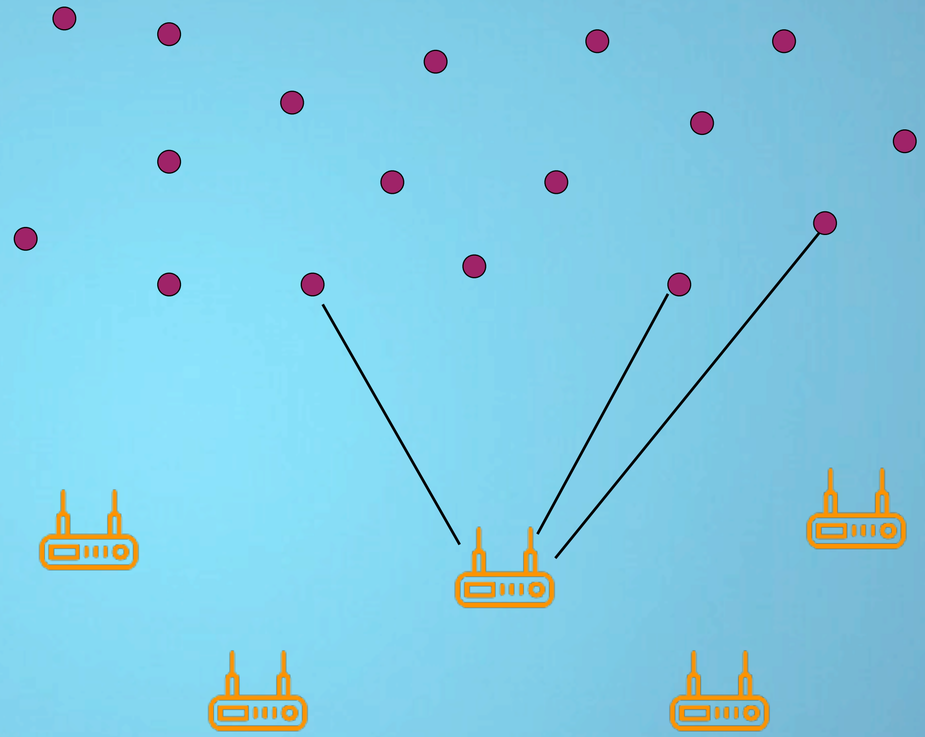
Application layer

Detect double spending, run consensus



Validators layer

Ensure validator uptime, withstand Byzantine behaviour



Interconnect layer

Correlated failures and attacks
Are validators up? From whose perspective?

Real-world resilience risks in blockchains

- Censorship / validators offline or slow ⇒ unavailability

Real-world resilience risks in blockchains

- Censorship / validators offline or slow ⇒ unavailability

Penalties

So far we have considered perfectly well-behaved validators, but what about validators that do not make timely head, source and target votes or do so slowly?

The penalties for missing the target and source votes are equal to the rewards the attestor would have received had they submitted them. This means that instead of having the reward

SLASHING

Slashing is a more severe action that results in the forceful removal of a validator from the network and an associated loss of their staked ether. There are three ways a validator can be

Nyle zoning

✓ **Challenge 1:**
Open membership resilient to Sybil attacks (globally < ⅓ Byzantine validators)

Solution:
Use existing* proof-of-work approaches

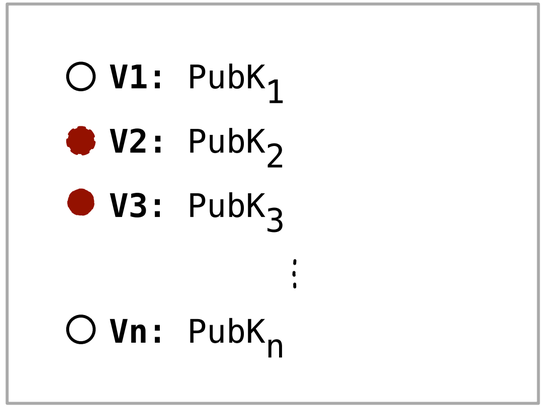
* Eleftherios Kokoris-Kogias, Philipp Jovanovic, Nicolas Gailly, Ismail Khoffi, Linus Gasser, and Bryan Ford. Enhancing Bitcoin Security and Performance with Strong Consistency via Collective Signing. In *Proceedings of the 25th USENIX Conference on Security Symposium*, 2016.

Nyle zoning

✓ **Challenge 1:**
Open membership resilient to Sybil attacks (globally < ⅓ Byzantine validators)

Solution:
Use existing* proof-of-work approaches

Global membership (<⅓ Byzantine)



Honest validator

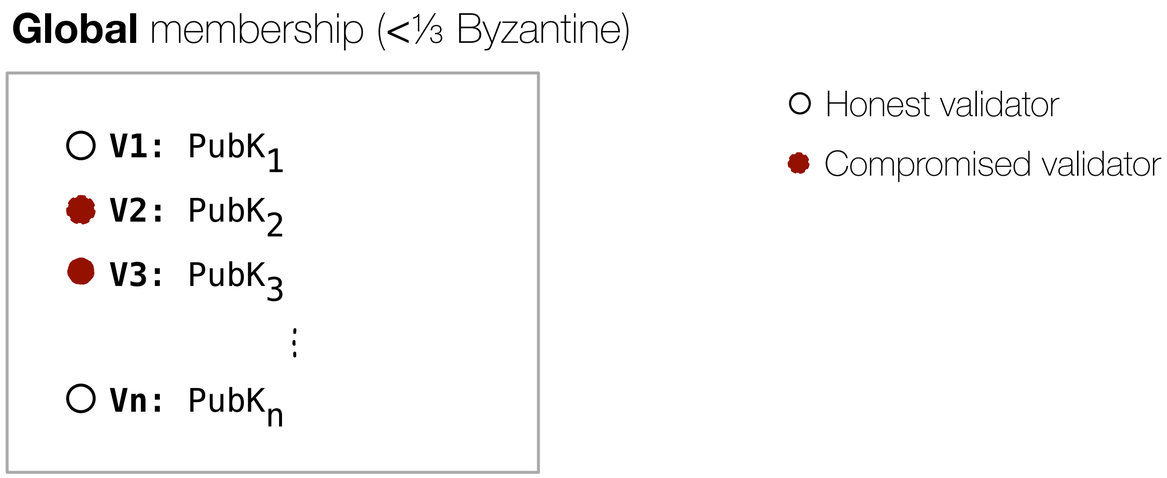
Compromised validator

* Eleftherios Kokoris-Kogias, Philipp Jovanovic, Nicolas Gailly, Ismail Khoffi, Linus Gasser, and Bryan Ford. Enhancing Bitcoin Security and Performance with Strong Consistency via Collective Signing. In *Proceedings of the 25th USENIX Conference on Security Symposium*, 2016.

Nyle zoning

✓ **Challenge 1:**
Open membership resilient to Sybil attacks (globally < ⅓ Byzantine validators)

Solution:
Use existing* proof-of-work approaches



X **Challenge 2:**
Validator joins two disjoint zones

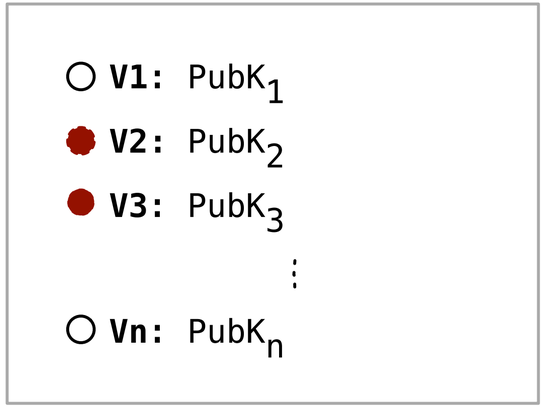
* Eleftherios Kokoris-Kogias, Philipp Jovanovic, Nicolas Gailly, Ismail Khoffi, Linus Gasser, and Bryan Ford. Enhancing Bitcoin Security and Performance with Strong Consistency via Collective Signing. In *Proceedings of the 25th USENIX Conference on Security Symposium*, 2016.

Nyle zoning

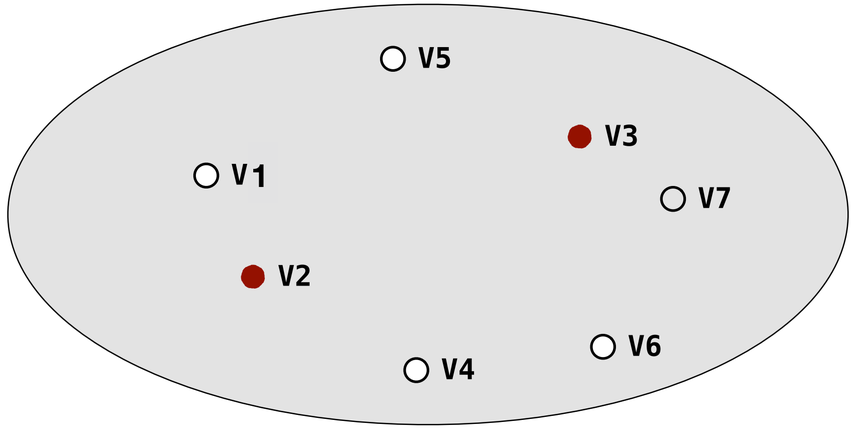
✓ **Challenge 1:**
Open membership resilient to Sybil attacks (globally < ⅓ Byzantine validators)

Solution:
Use existing* proof-of-work approaches

Global membership (<⅓ Byzantine)



X **Challenge 2:**
Validator joins two disjoint zones



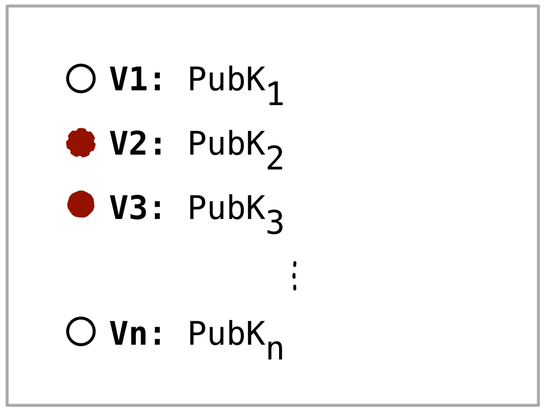
* Eleftherios Kokoris-Kogias, Philipp Jovanovic, Nicolas Gailly, Ismail Khoffi, Linus Gasser, and Bryan Ford. Enhancing Bitcoin Security and Performance with Strong Consistency via Collective Signing. In *Proceedings of the 25th USENIX Conference on Security Symposium*, 2016.

Nyle zoning

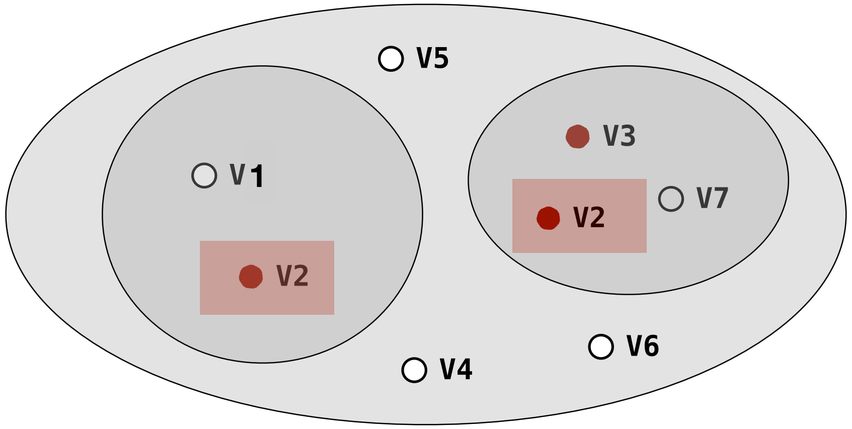
✓ **Challenge 1:**
Open membership resilient to Sybil attacks (globally < ⅓ Byzantine validators)

Solution:
Use existing* proof-of-work approaches

Global membership (<⅓ Byzantine)



✗ **Challenge 2:**
Validator joins two disjoint zones



* Eleftherios Kokoris-Kogias, Philipp Jovanovic, Nicolas Gailly, Ismail Khoffi, Linus Gasser, and Bryan Ford. Enhancing Bitcoin Security and Performance with Strong Consistency via Collective Signing. In *Proceedings of the 25th USENIX Conference on Security Symposium*, 2016.

Nyle zoning

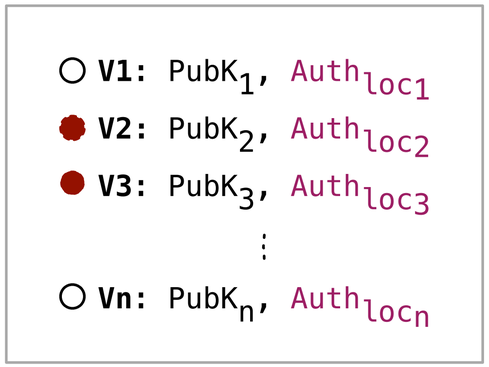
✓ **Challenge 1:**
Open membership resilient to Sybil attacks (globally < ⅓ Byzantine validators)

Solution:
Use existing proof-of-work approaches

✓ **Challenge 2:**
Validator joins two disjoint zones

Solution:
Validators make a location claim

Global membership (<⅓ Byzantine)



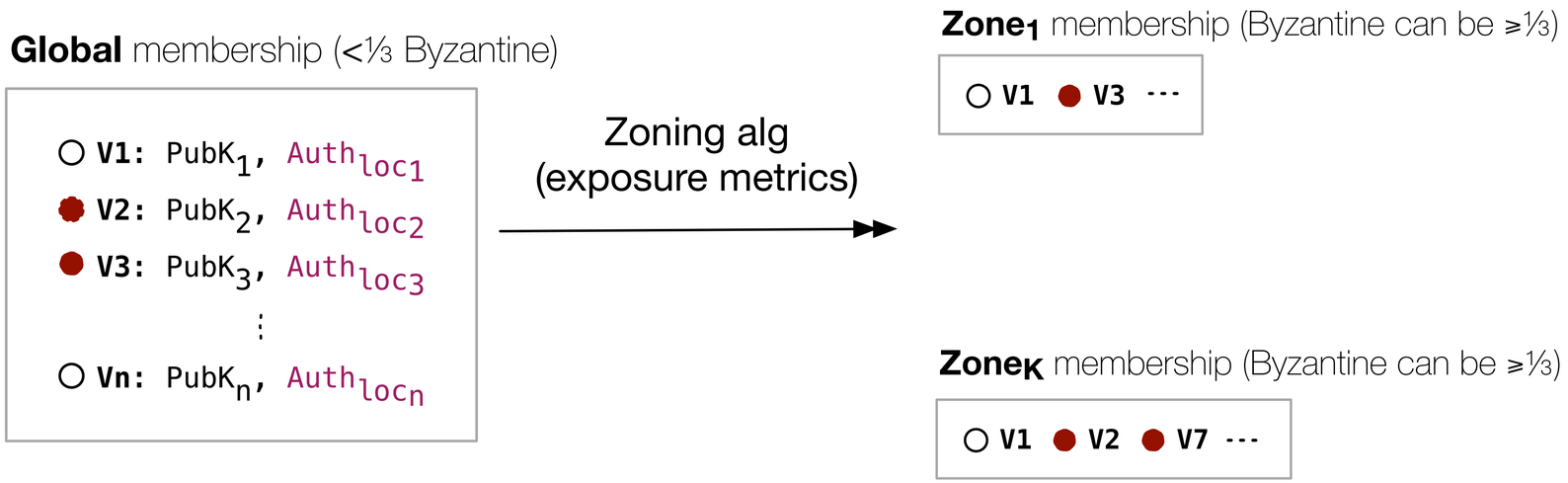
Nyle zoning

√ **Challenge 1:**
Open membership resilient to Sybil attacks (globally < ⅓ Byzantine validators)

Solution:
Use existing proof-of-work approaches

√ **Challenge 2:**
Validator joins two disjoint zones

Solution:
Validators make a location claim



Nyle zoning

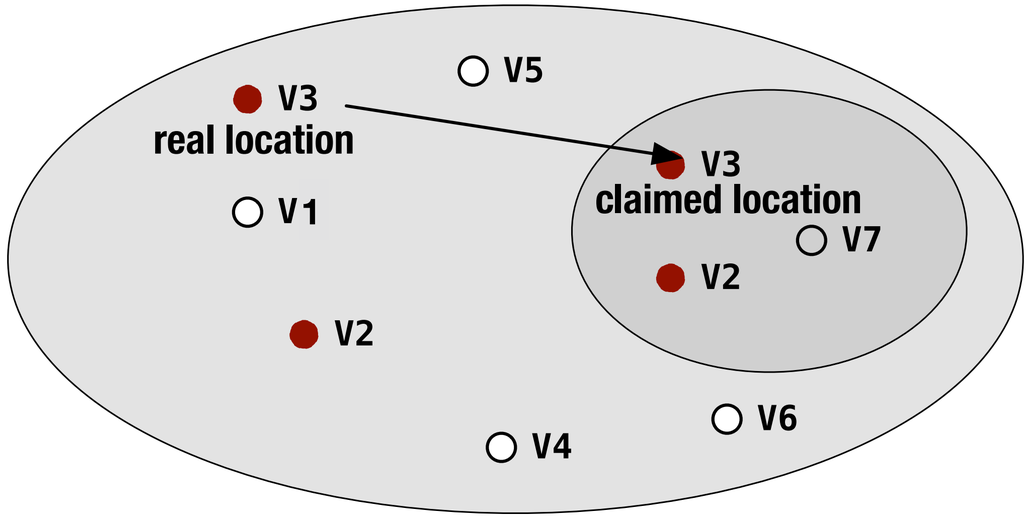
✓ **Challenge 1:**
Open membership resilient to Sybil attacks (globally < ⅓ Byzantine validators)

Solution:
Use existing proof-of-work approaches

✓ **Challenge 2:**
Validator joins two disjoint zones

Solution:
Validators make a location claim

X **Challenge 3:**
Compromised validator makes bogus location claim



Nyle zoning

√ **Challenge 1:**
Open membership resilient to Sybil attacks (globally < ⅓ Byzantine validators)

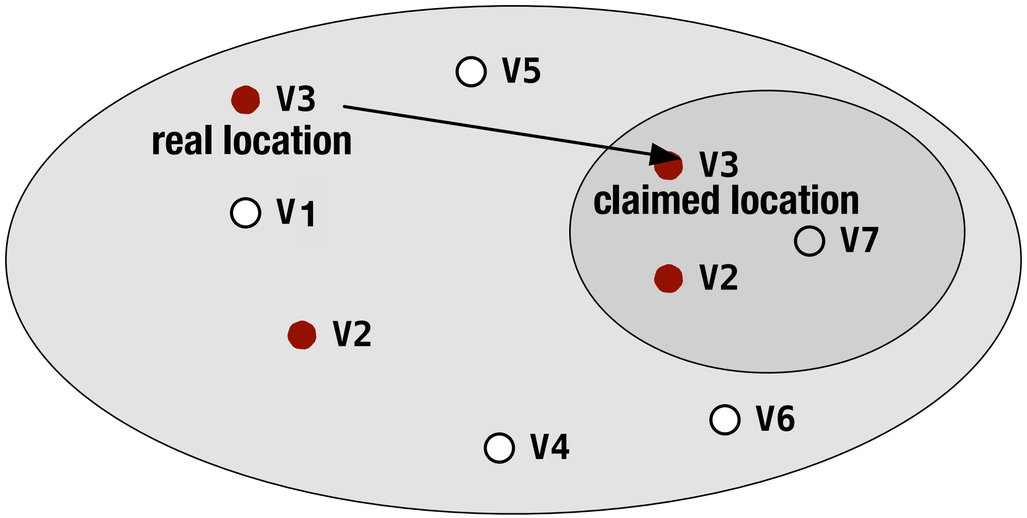
Solution:
Use existing proof-of-work approaches

√ **Challenge 2:**
Validator joins two disjoint zones

Solution:
Validators make a location claim

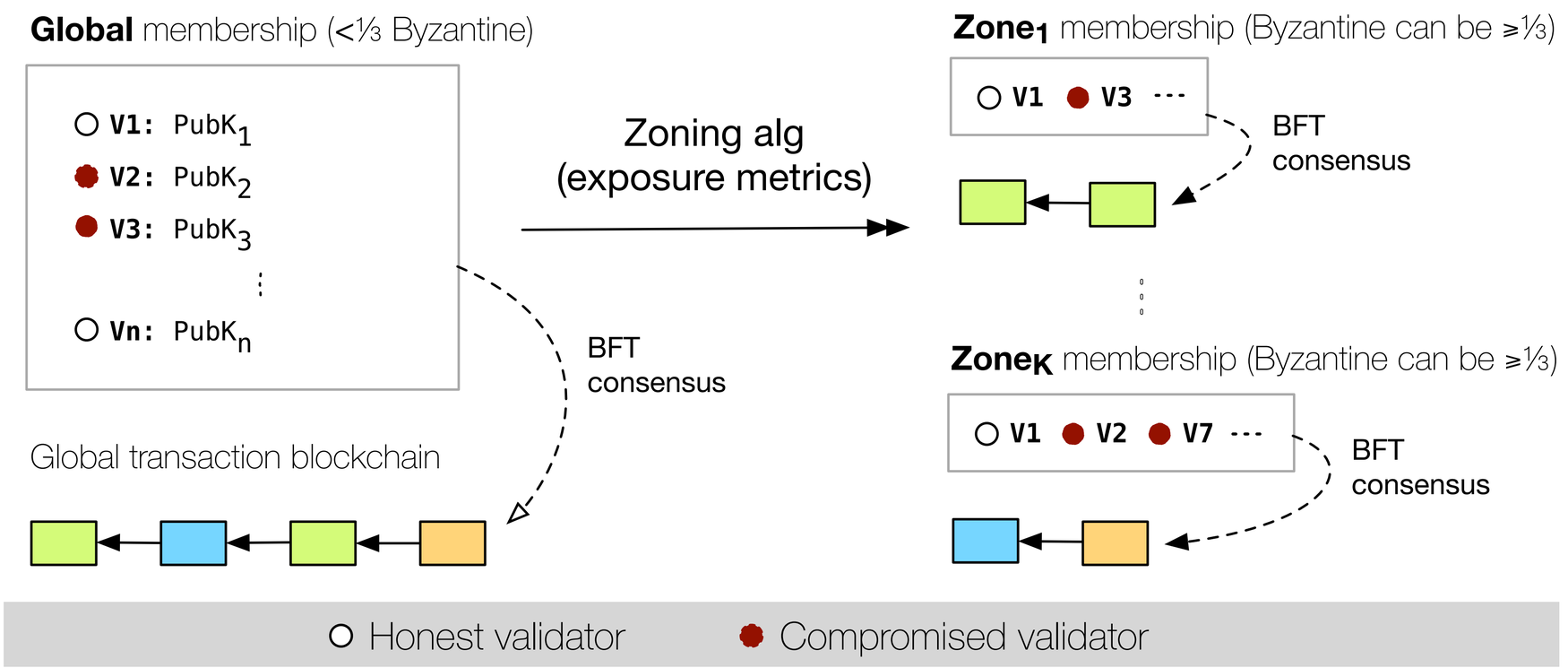
√ **Challenge 3:**
Malicious validator makes wrong location claim

Solution:
Validate location claims via existing mechanisms*



* Katharina Kohls and Claudia Diaz. VerLoc: Verifiable localization in decentralized systems. In *USENIX Security Symposium (USENIX Security)*, 2022

Nyle zoning


Global membership (<⅓ Byzantine)
V1: PubK1
V2: PubK2
V3: PubK3
Vn: PubKn
Zoning alg
(exposure metrics)
BFT
consensus
Global transaction blockchain
Zone1 membership (Byzantine can be ⩾⅓)
V1 V3 ---
BFT
consensus
ZoneK membership (Byzantine can be ⩾⅓)
V1 V2 V7 ---
BFT
consensus
Honest validator
Compromised validator

Nyle transaction validation security

#	Participant honest(✓) / compromised(×) / unknown(∗)				Nyle property guarantee yes(✓) / no(×) / N/A [reason]						
					Safety			Liveness*			Exposure
	Sender	Recv	CLZ	SEZ	Sender	Recv.	Zones	Sender	Recv.	Zones	
1.	✓	*	✓	✓	✓	✓	✓	✓	✓	✓	SEZ
2.				×		✓[S2]			✓[L1]		Global
3.	✓	*	×	✓	✓[S1]	✓	✓	✓[L2]	✓	✓[L2]	SEZ
4.				×		✓[S2]			✓[L1,L2]		Global
5.	×	*	✓	✓	N/A	✓	✓	N/A	✓	✓	SEZ
6.				×		✓[S4]			✓[L1]		Global
7.	×	*	×	✓	N/A	✓[S3]	✓[S3]	N/A	✓	✓[L1,L3]	SEZ
8.				×		✓[S3]			✓[L1]		Global

Reason ID	Description
S1	The sender can forward an already validated transaction to cancel CLZ history rewrite.
S2	The transaction should be accepted, the receiver detects the bogus reject of a compromised SEZ.
S3	Zones and receiver notified of double spending by the Global zone.
S4	The transaction must not be accepted, the receiver detects the bogus accept of a compromised SEZ.
L1	The Global zone eventually replies with an authentic validation.
L2	The sender replaces a censoring CLZ via a CLZ transfer (Section 3.6).
L3	The zones contact the Global zone to intervene if the CLZ is unresponsive.

Binary consensus using common core

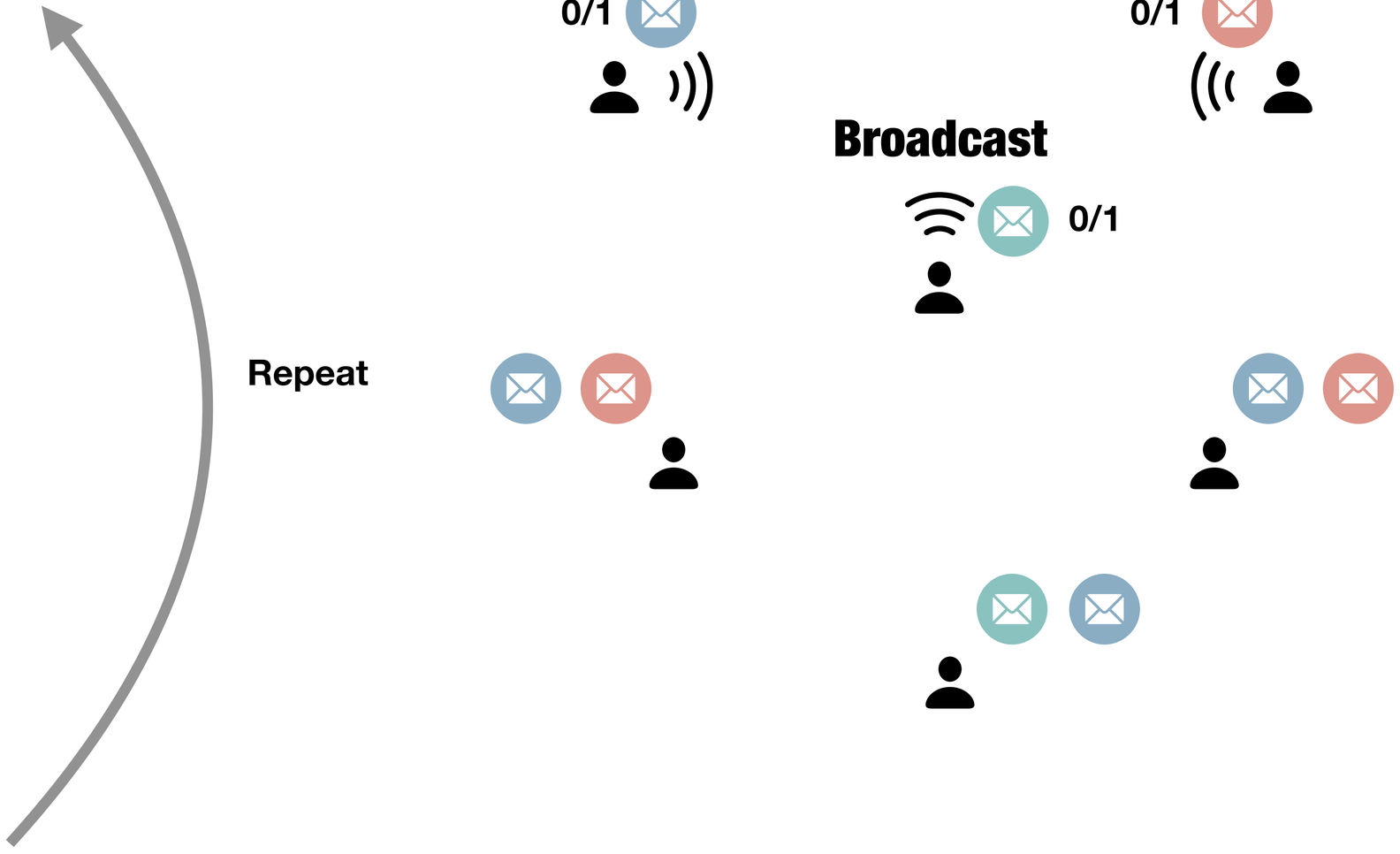
Exchange of information

Voting

Wait for N-f

Can I decide?
If not, how do I continue?

Observe a majority vote?



Consensus using reliable broadcast

Agreement on arbitrary values

Cannot use majority vote

Use reliable broadcast

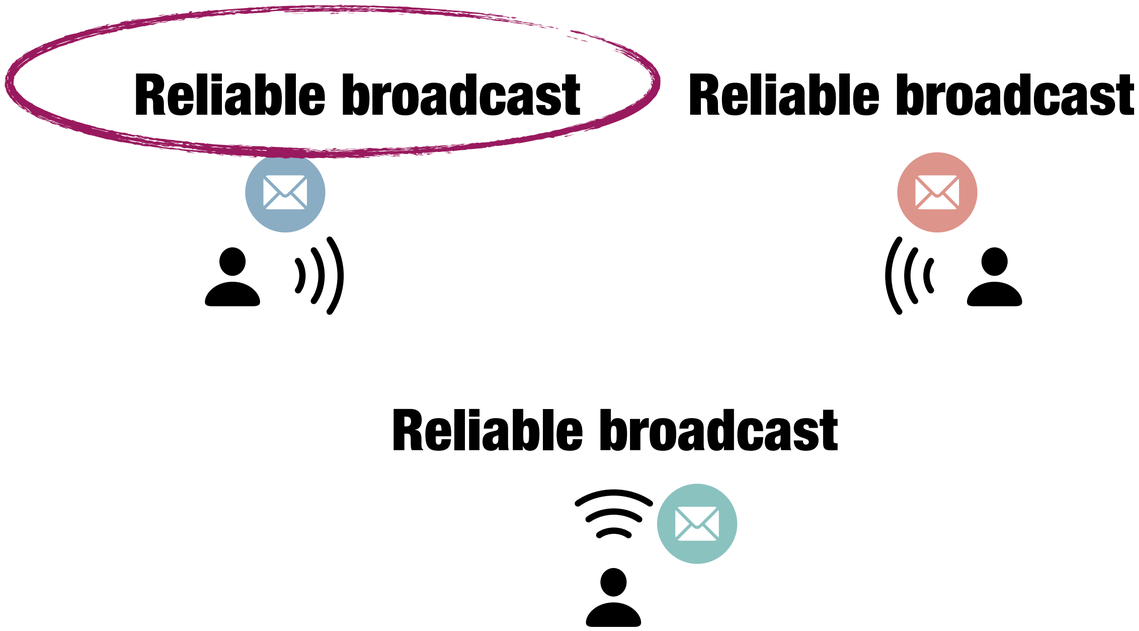
Eventually all live nodes receive the same proposals. But when?

Different signals possible

A multi-valued common coin
N blocks of binary consensus

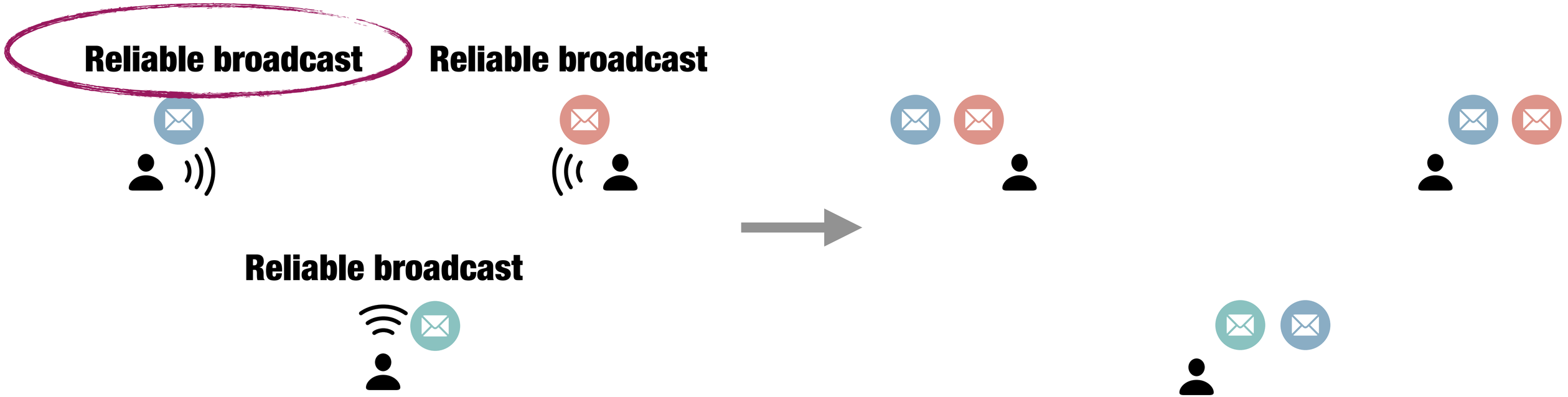
Consensus with common coins and reliable broadcast

Validity

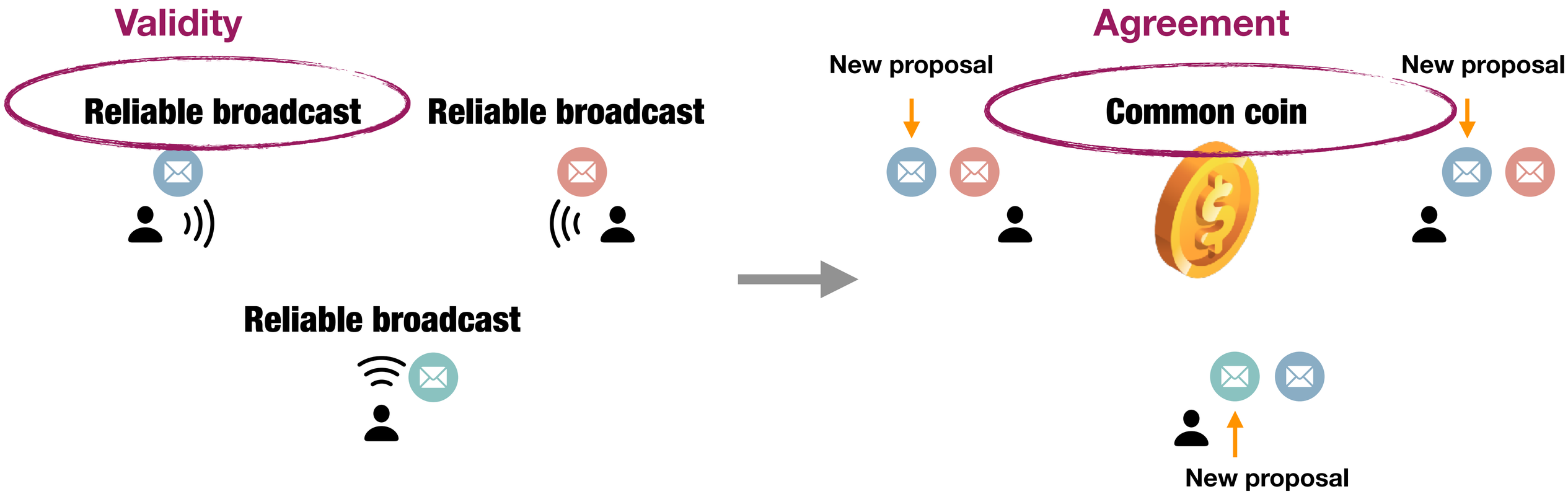


Consensus with common coins and reliable broadcast

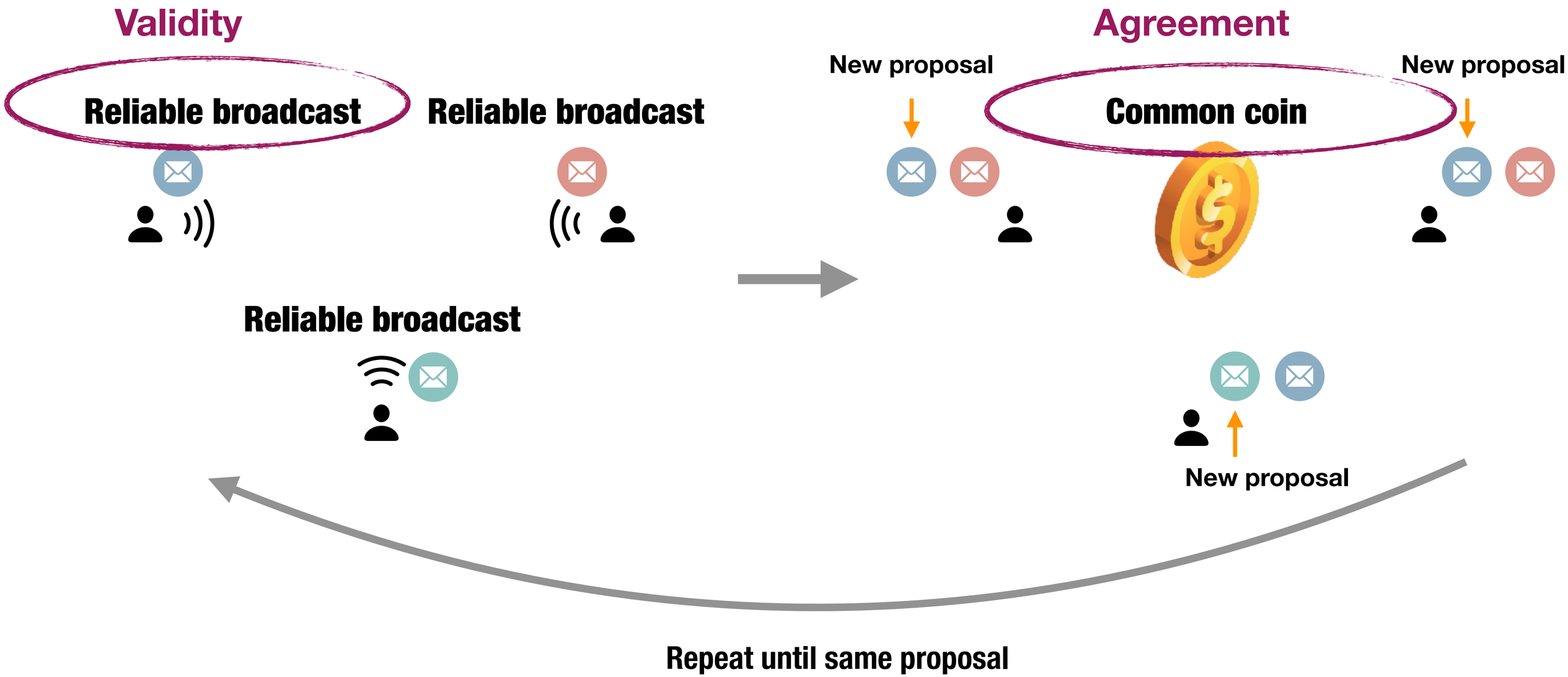
Validity



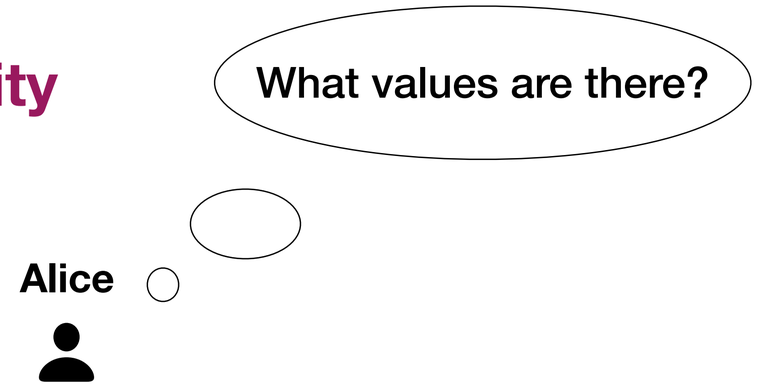
Consensus with common coins and reliable broadcast



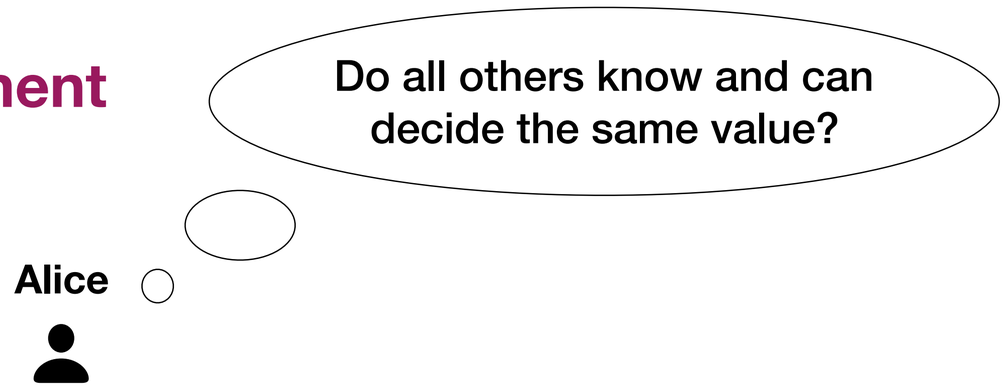
Consensus with common coins and reliable broadcast



Validity

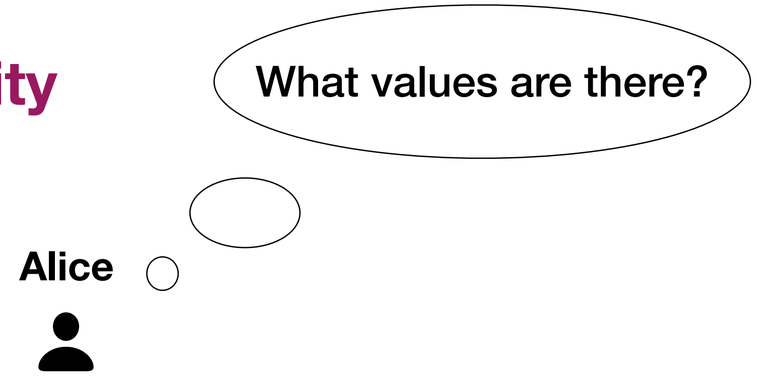


Agreement

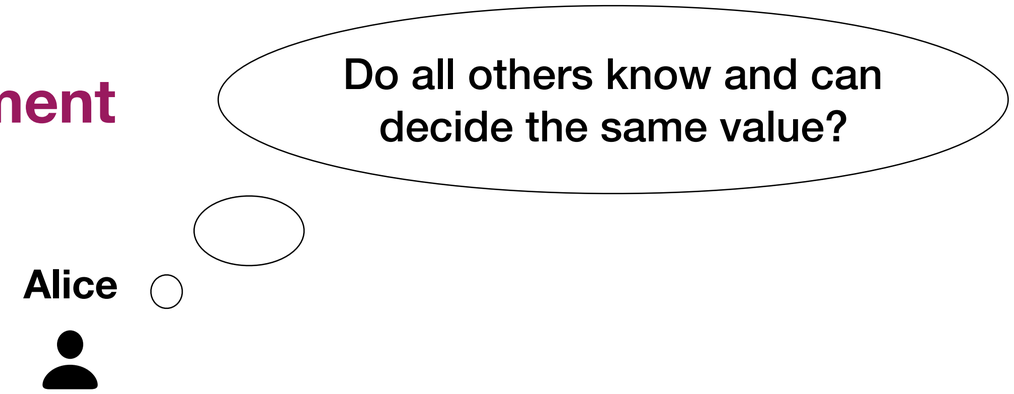


QSC insights

Validity

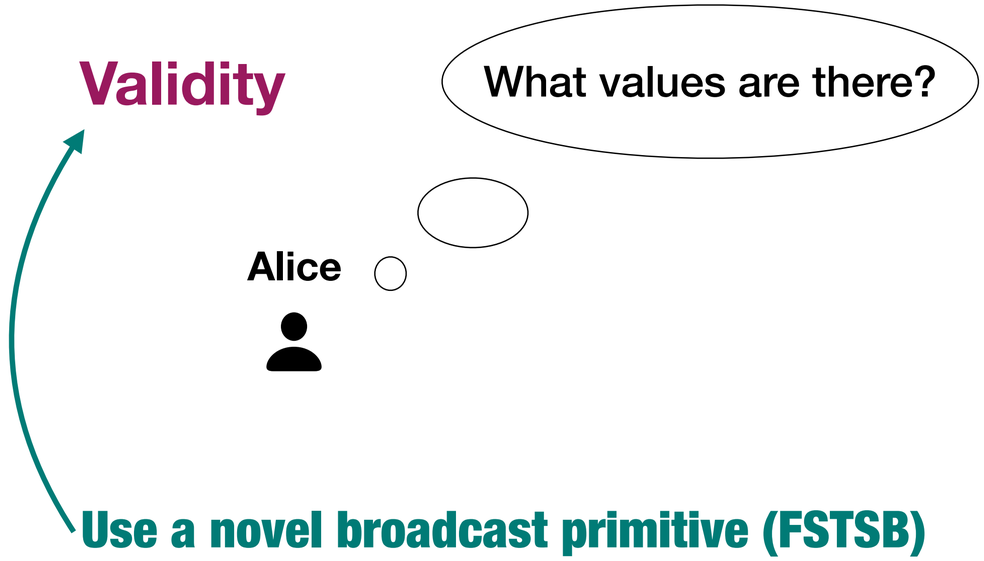


Agreement



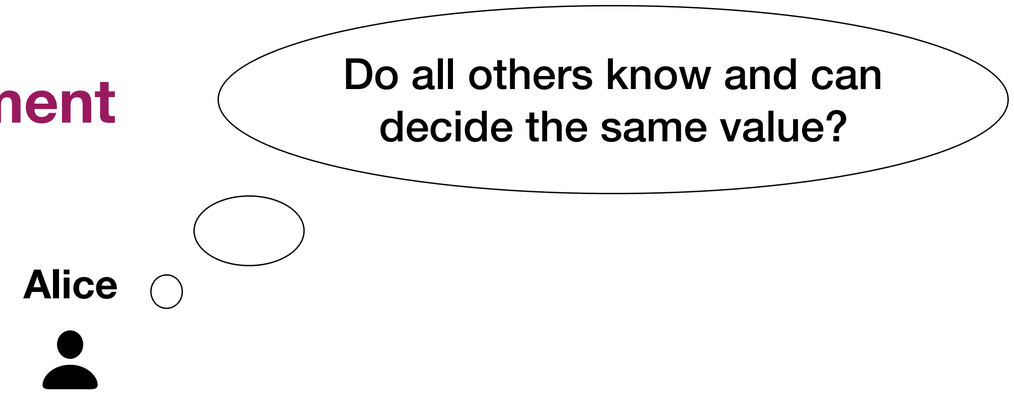
QSC insights

Validity



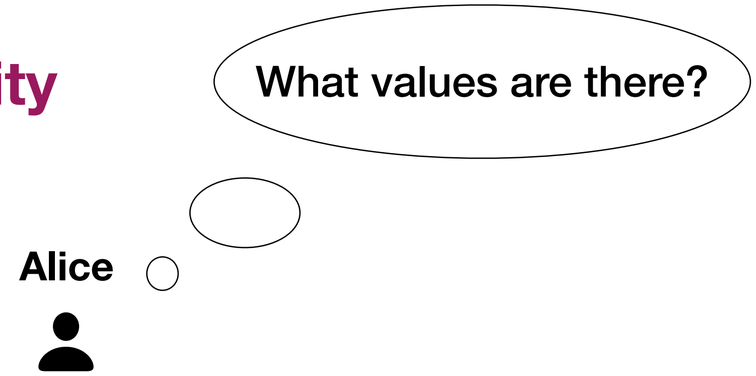
Use a novel broadcast primitive (FSTSB)

Agreement

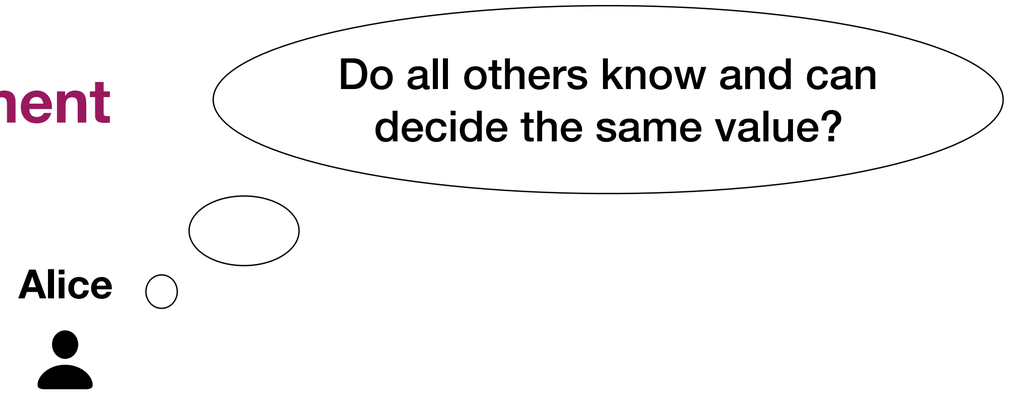


QSC insights

Validity



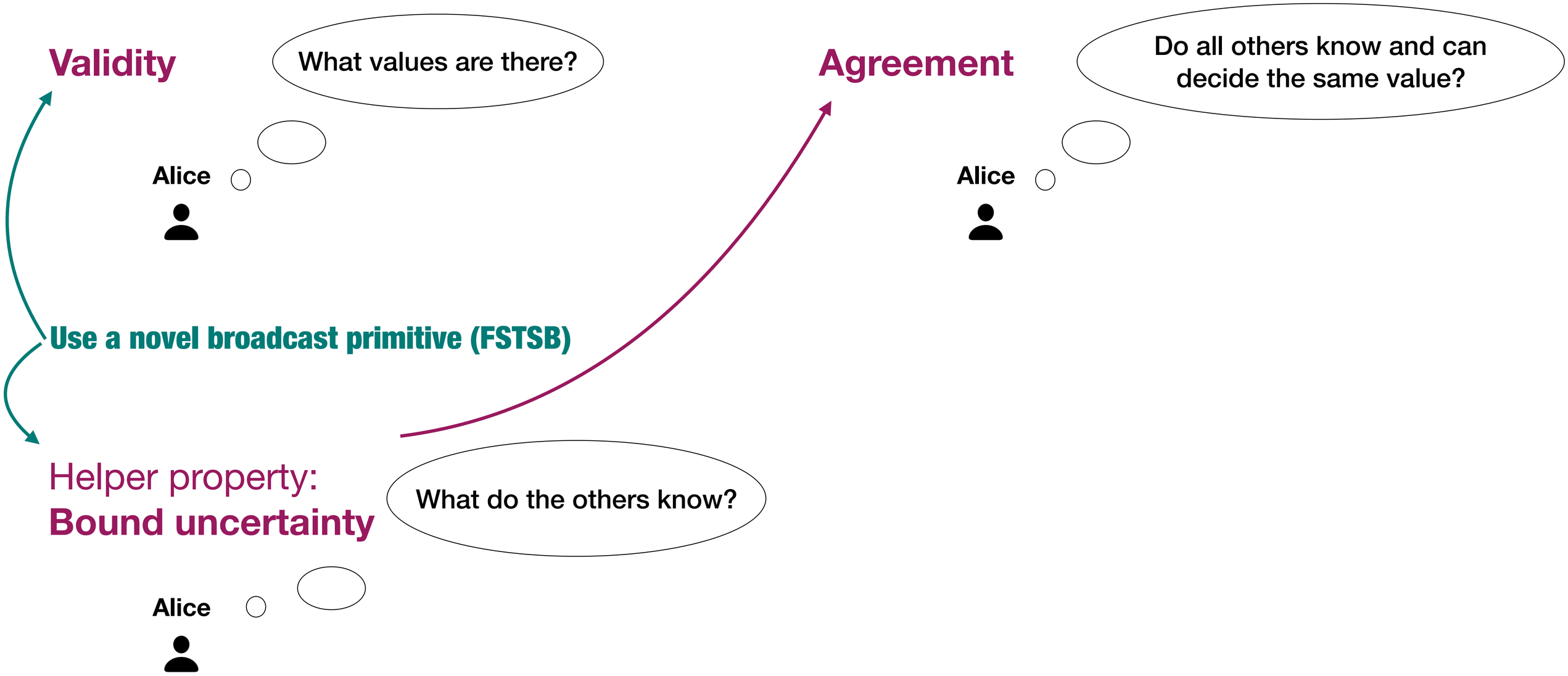
Agreement



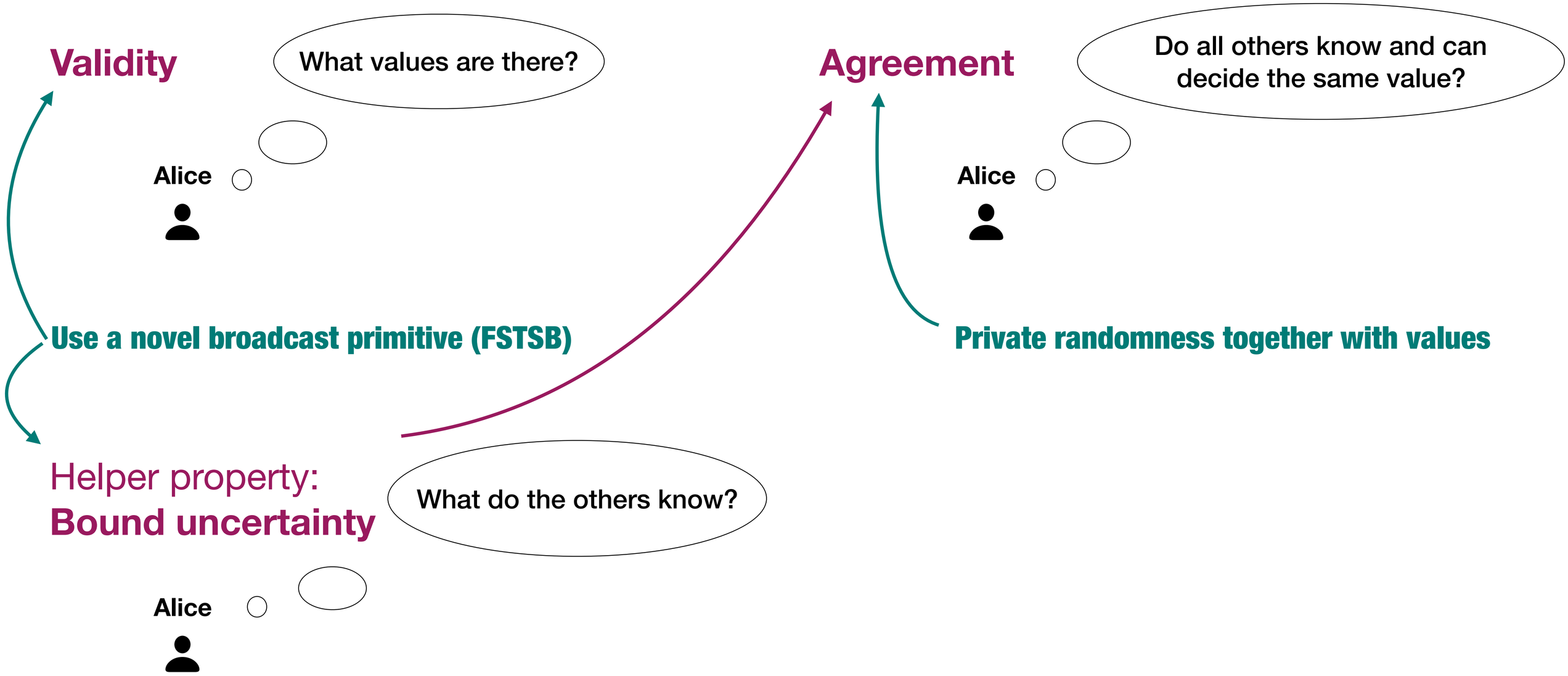
Use a novel broadcast primitive (FSTSB)

Helper property:
Bound uncertainty

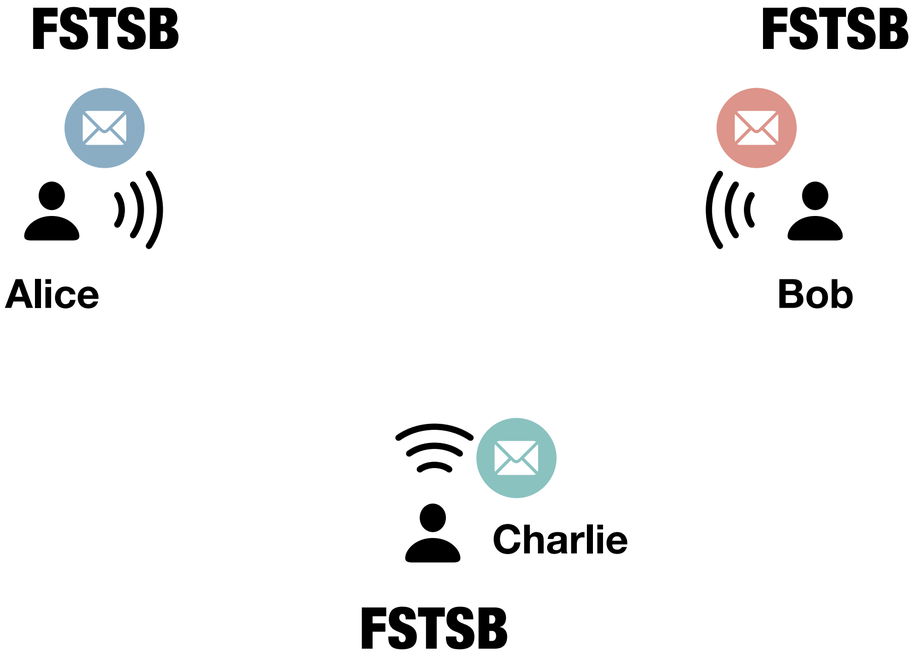
QSC insights



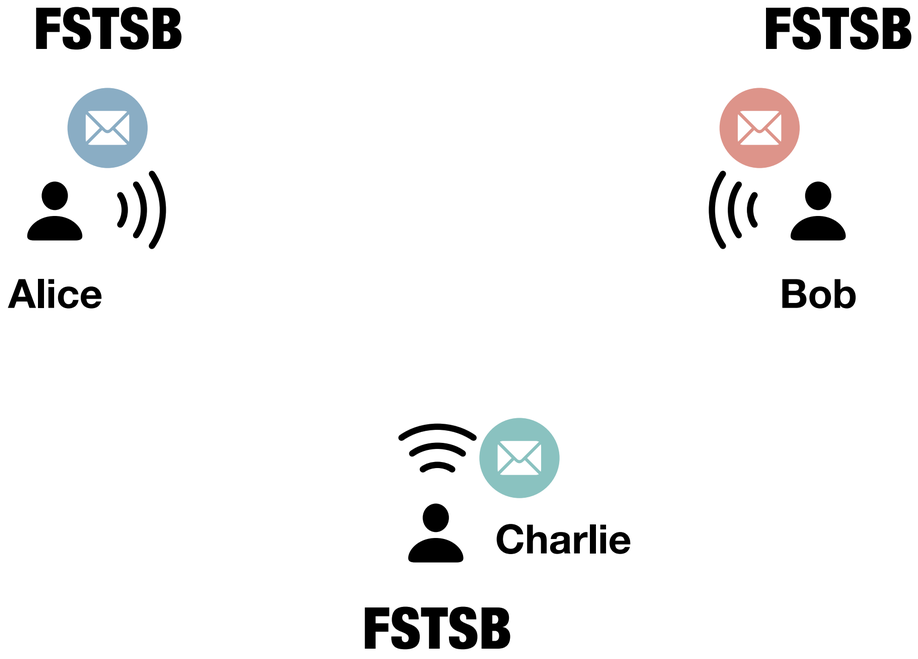
QSC insights



Validity



Validity



- No other proposal is ever introduced

Validity

Alice

Bob

FSTSB

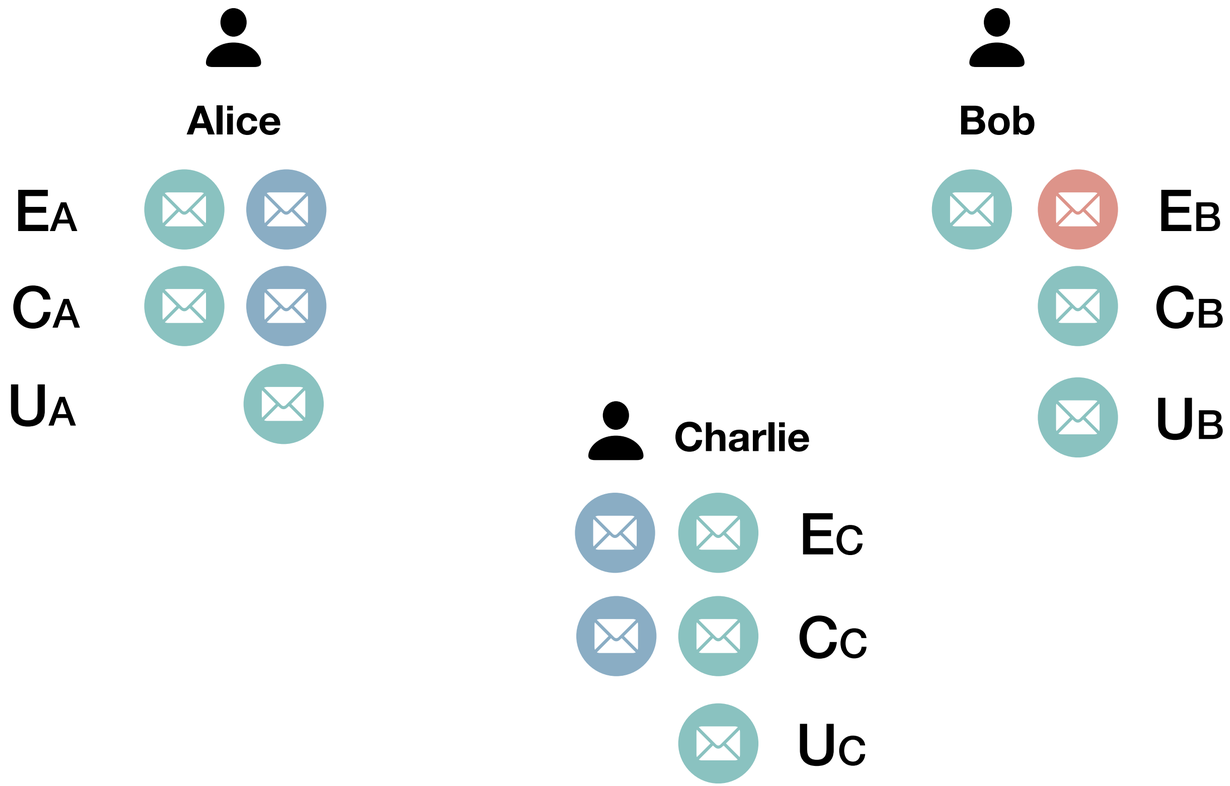
- No other proposal is ever introduced
- Though random values change

Bounding uncertainty by approximating others' knowledge

FSTSB ⇒ sets E,C,U

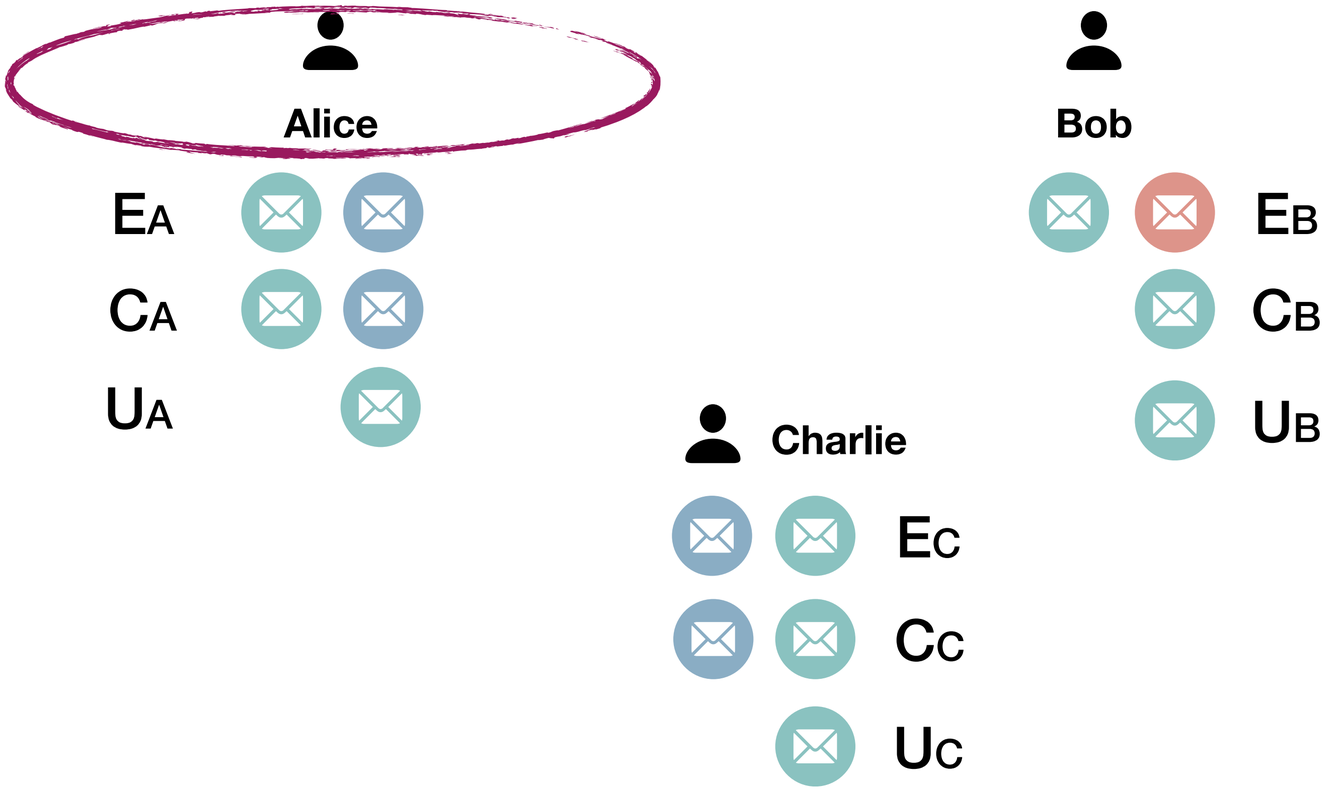
Bounding uncertainty by approximating others' knowledge

FSTSB ⇒ sets E,C,U



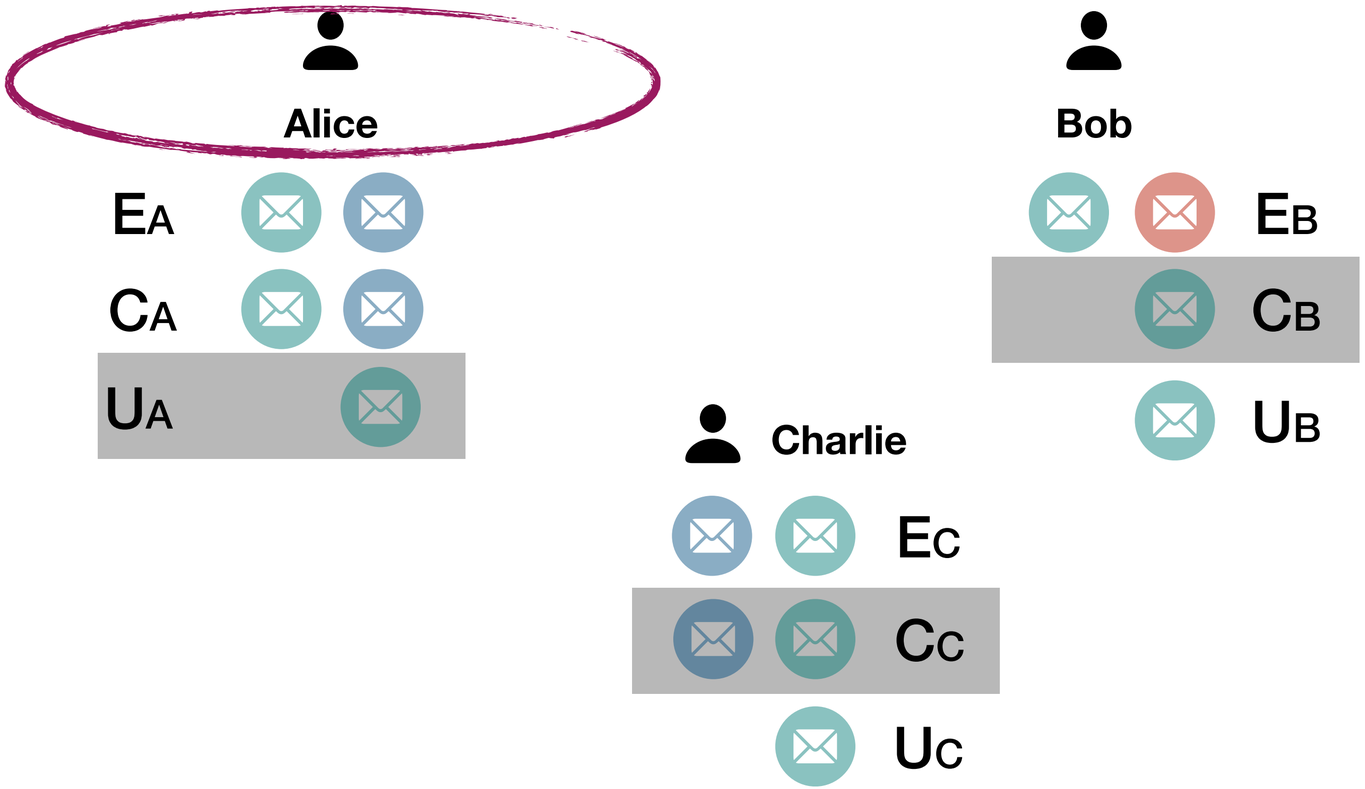
Bounding uncertainty by approximating others' knowledge

FSTSB ⇒ sets E,C,U



Bounding uncertainty by approximating others' knowledge

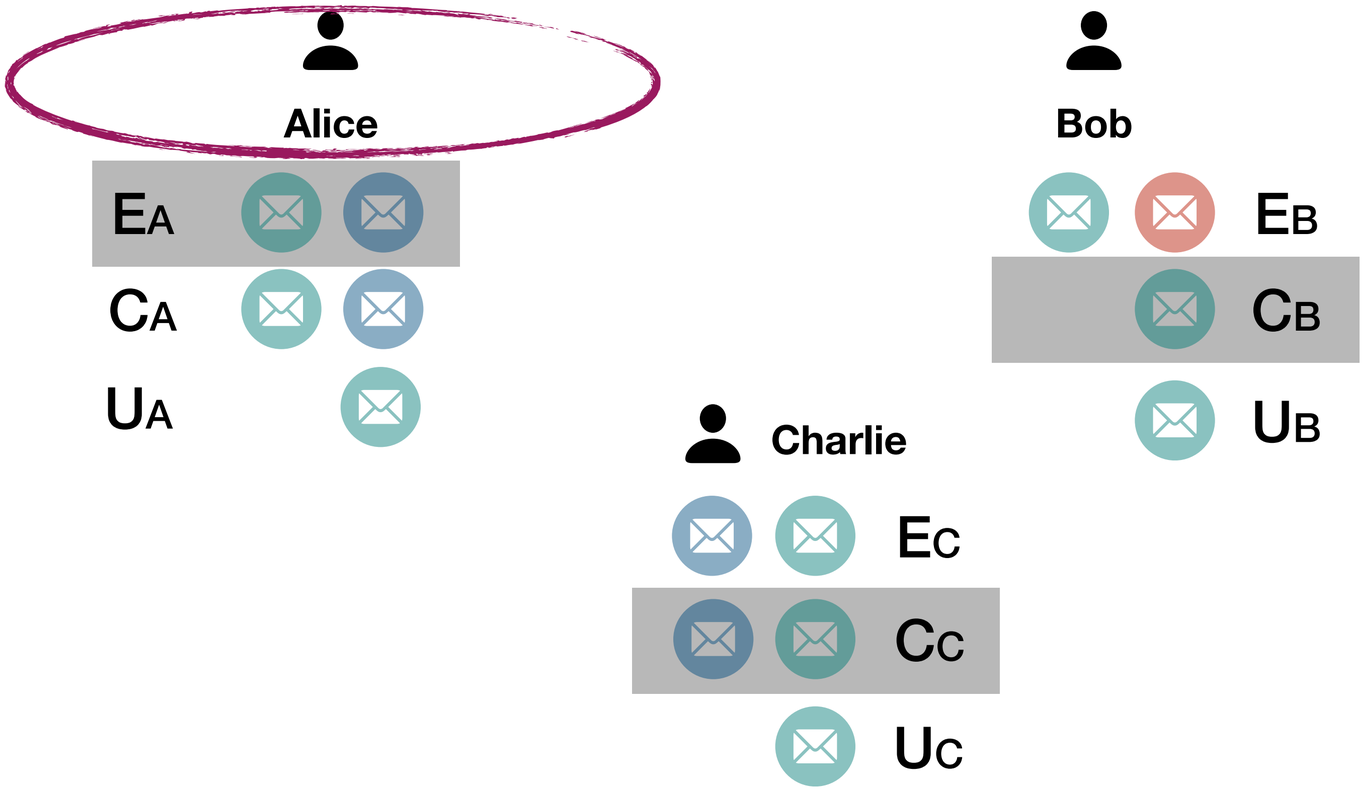
FSTSB ⇒ sets E,C,U



$C_{\text{any-other}} \supseteq U_{\text{Alice}}$

Bounding uncertainty by approximating others' knowledge

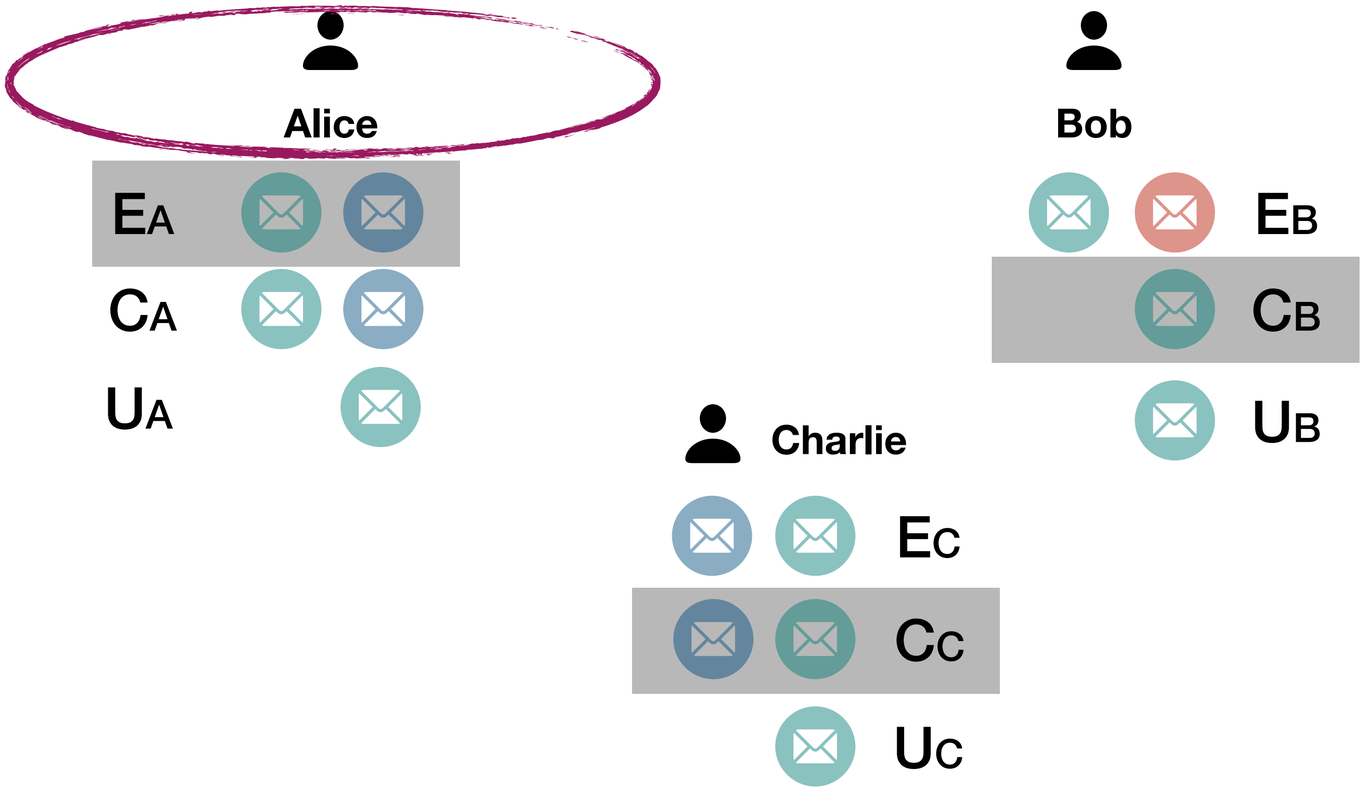
FSTSB ⇒ sets E,C,U



$C_{\text{any-other}} \supseteq U_{\text{Alice}}$

Bounding uncertainty by approximating others' knowledge

FSTSB ⇒ sets E,C,U



Bounding uncertainty by approximating others' knowledge

FSTSB ⇒ sets E,C,U

Alice

Bounding uncertainty by approximating others' knowledge

FSTSB ⇒ sets E,C,U

Alice

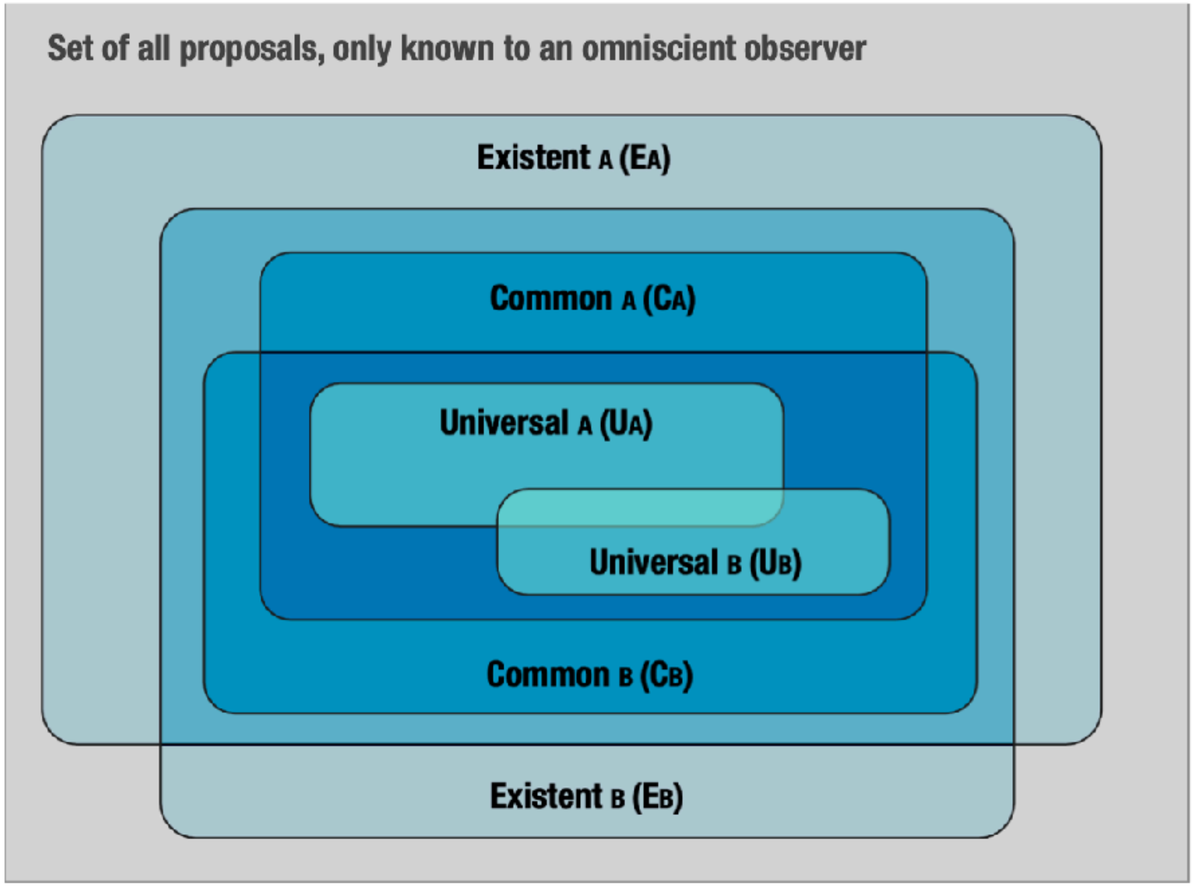
$$\mathbf{E}_{\text{Alice}} \supseteq \mathbf{C}_{\text{any-other}} \supseteq \mathbf{U}_{\text{Alice}}$$

Bounding uncertainty by approximating others' knowledge

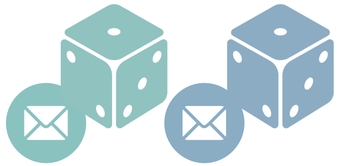
FSTSB ⇒ sets E,C,U

Alice

$E_{Alice} \supseteq C_{any\text{-}other} \supseteq U_{Alice}$



Agreement



$\mathbf{E}_{\text{Alice}} \supseteq \mathbf{C}_{\text{any-other}} \supseteq \mathbf{U}_{\text{Alice}}$

Agreement

E Alice ⊇ **C** any-other ⊇ **U** Alice

Agreement

$$\mathbf{E}_{\text{Alice}} \supseteq \mathbf{C}_{\text{any-other}} \supseteq \mathbf{U}_{\text{Alice}}$$

Highest priority $\overset{?}{=}$ Highest priority

Agreement

$$\mathbf{E}_{\text{Alice}} \supseteq \mathbf{C}_{\text{any-other}} \supseteq \mathbf{U}_{\text{Alice}}$$

Highest priority $\overset{?}{=}$ Highest priority

Y: Decide that value

Agreement

$$E_{\text{Alice}} \supseteq C_{\text{any-other}} \supseteq U_{\text{Alice}}$$

Highest priority $\overset{?}{=}$ Highest priority

Y: Decide that value

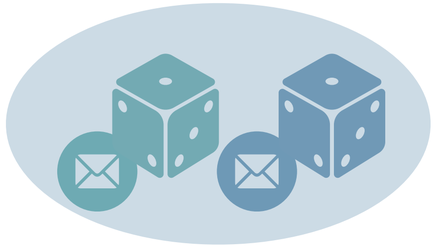
$$E_{\text{any-other}} \supseteq C_{\text{Alice}} \supseteq U_{\text{any-other}}$$

Agreement

$$E_{\text{Alice}} \supseteq C_{\text{any-other}} \supseteq U_{\text{Alice}}$$

Highest priority $\stackrel{?}{=}$ Highest priority

Y: Decide that value



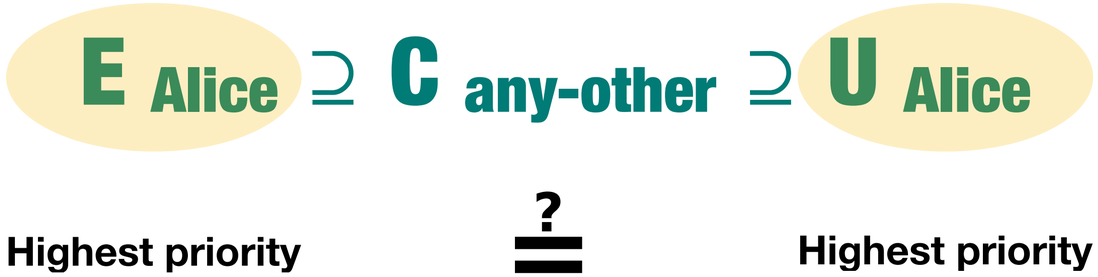
$$E_{\text{any-other}} \supseteq C_{\text{Alice}} \supseteq U_{\text{any-other}}$$

N: Continue with highest priority

Agreement

$$\mathbf{E}_{\text{Alice}} \supseteq \mathbf{C}_{\text{any-other}} \supseteq \mathbf{U}_{\text{Alice}}$$

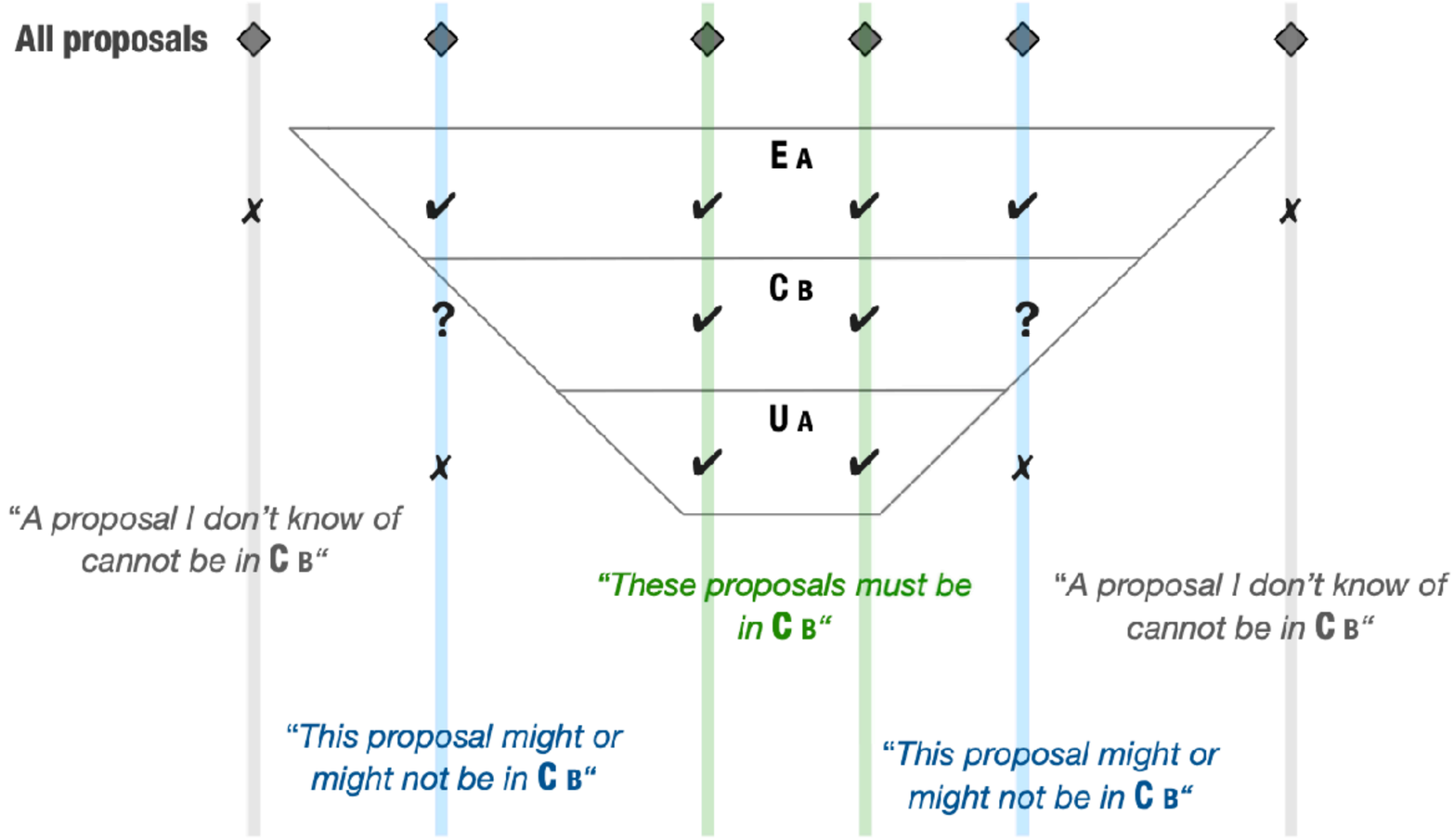
Agreement


E Alice
⊇
C any-other
⊇
U Alice
Highest priority
?
Highest priority

Agreement

$E_{Alice} \supseteq C_{any\text{-}other} \supseteq U_{Alice}$

Highest priority $\overset{?}{=}$ Highest priority



Agreement

Continue with highest priority

$E_{\text{Alice}} \supseteq C_{\text{any-other}} \supseteq U_{\text{Alice}}$

Highest priority $\overset{?}{=}$ Highest priority