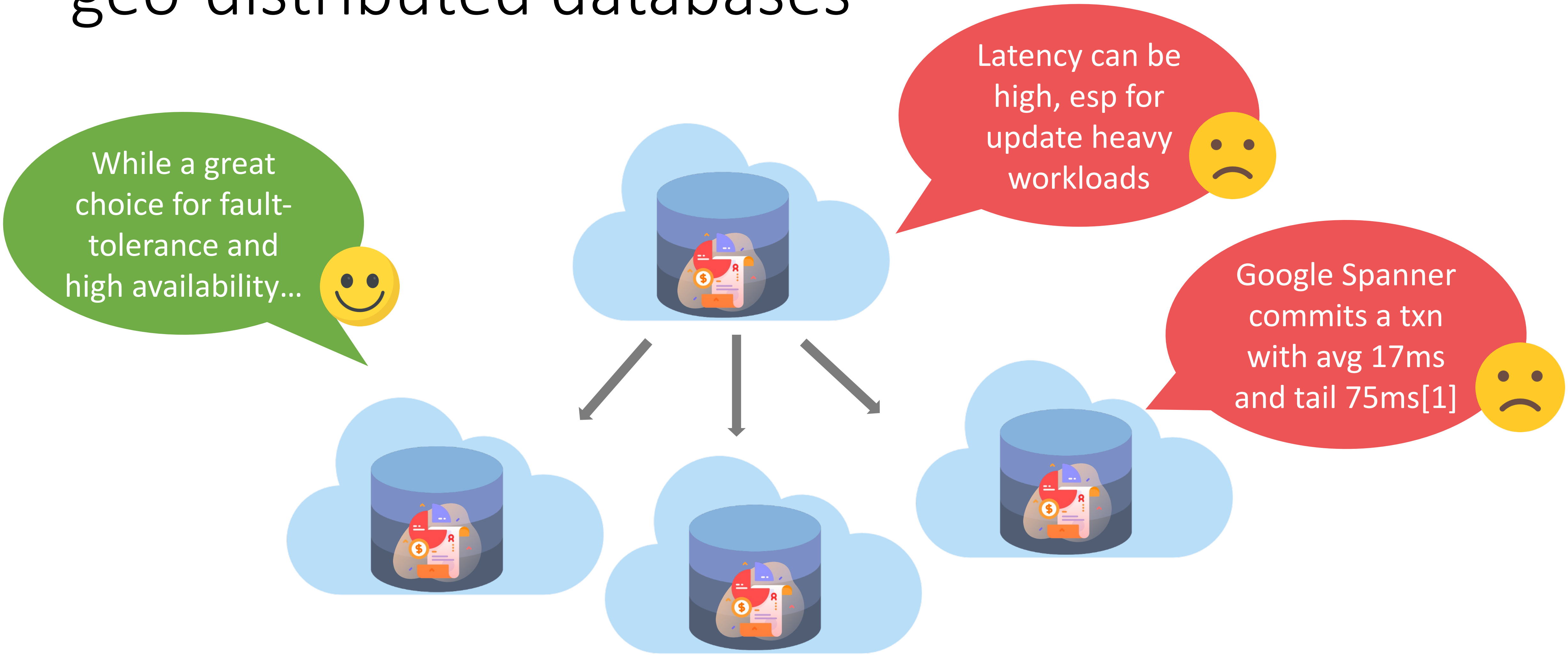
Samya: Geo-Distributed Data System for High Contention Data Aggregates

Sujaya Maiyya, Ishtiyaque Ahmad, Divyakant Agrawal, Amr El Abbadi

UC Santa Barbara

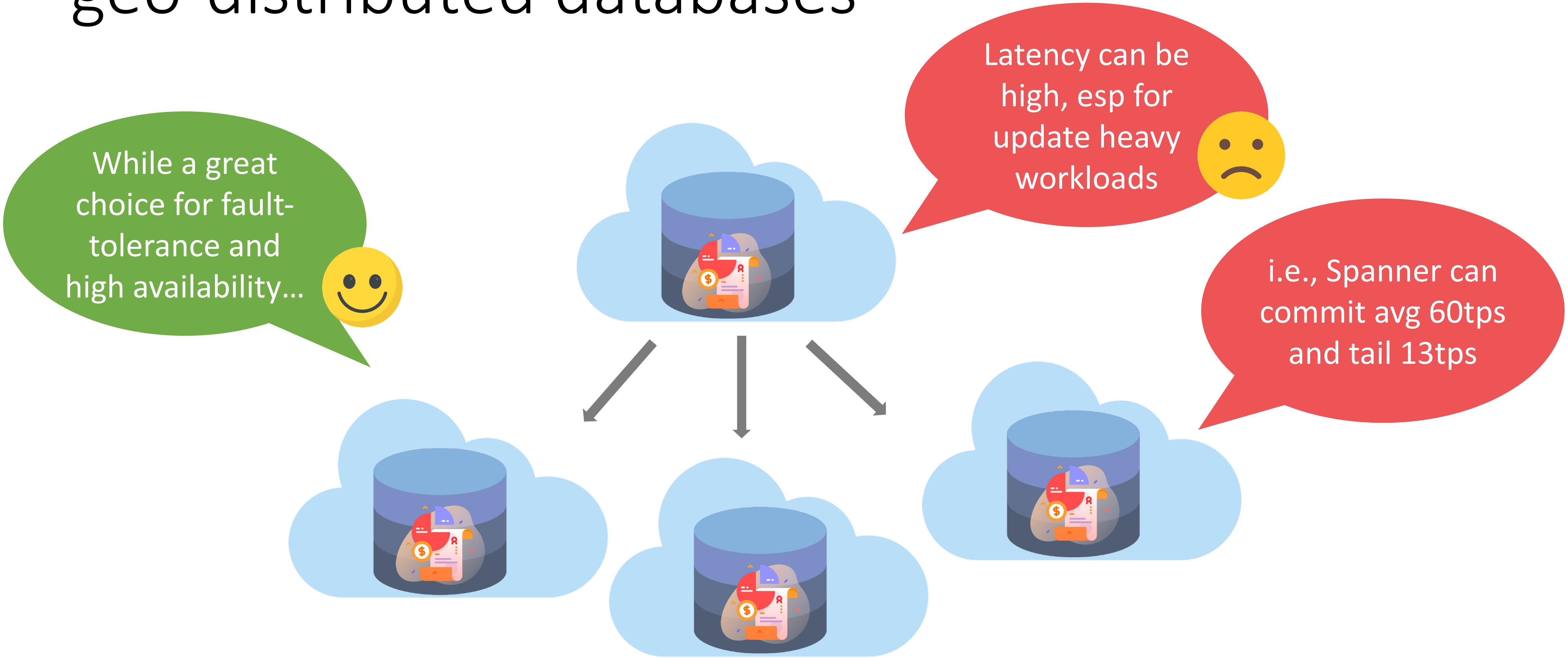
Today, we are in a world of geo-distributed databases

Today, we are in a world of geo-distributed databases



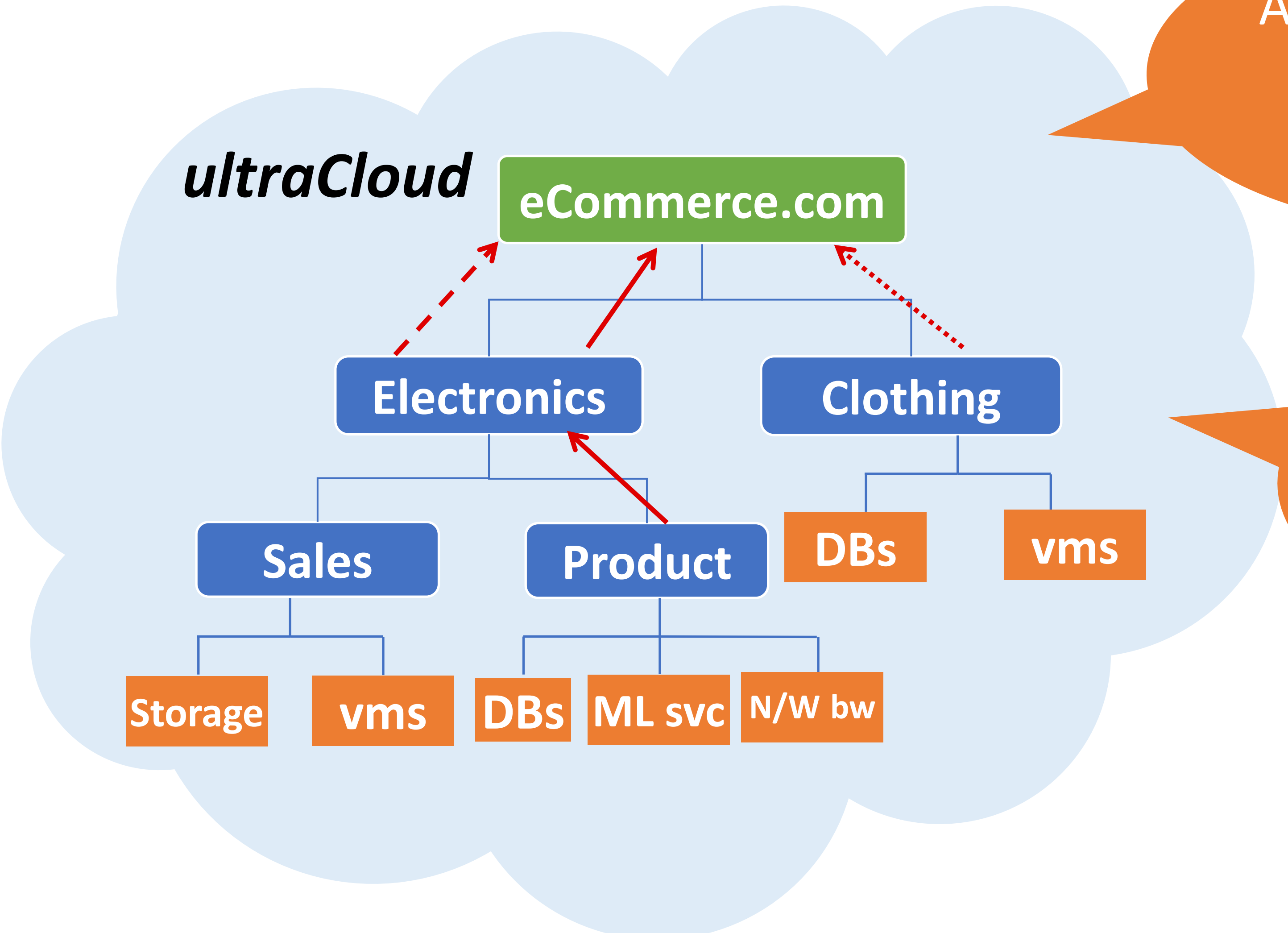
[1] J. C. Corbett et al. Spanner: Google’s globally distributed database. ACM Transactions on Computer Systems (TOCS), 2013.

Today, we are in a world of geo-distributed databases



[1] J. C. Corbett et al. Spanner: Google's globally distributed database. ACM Transactions on Computer Systems (TOCS), 2013.

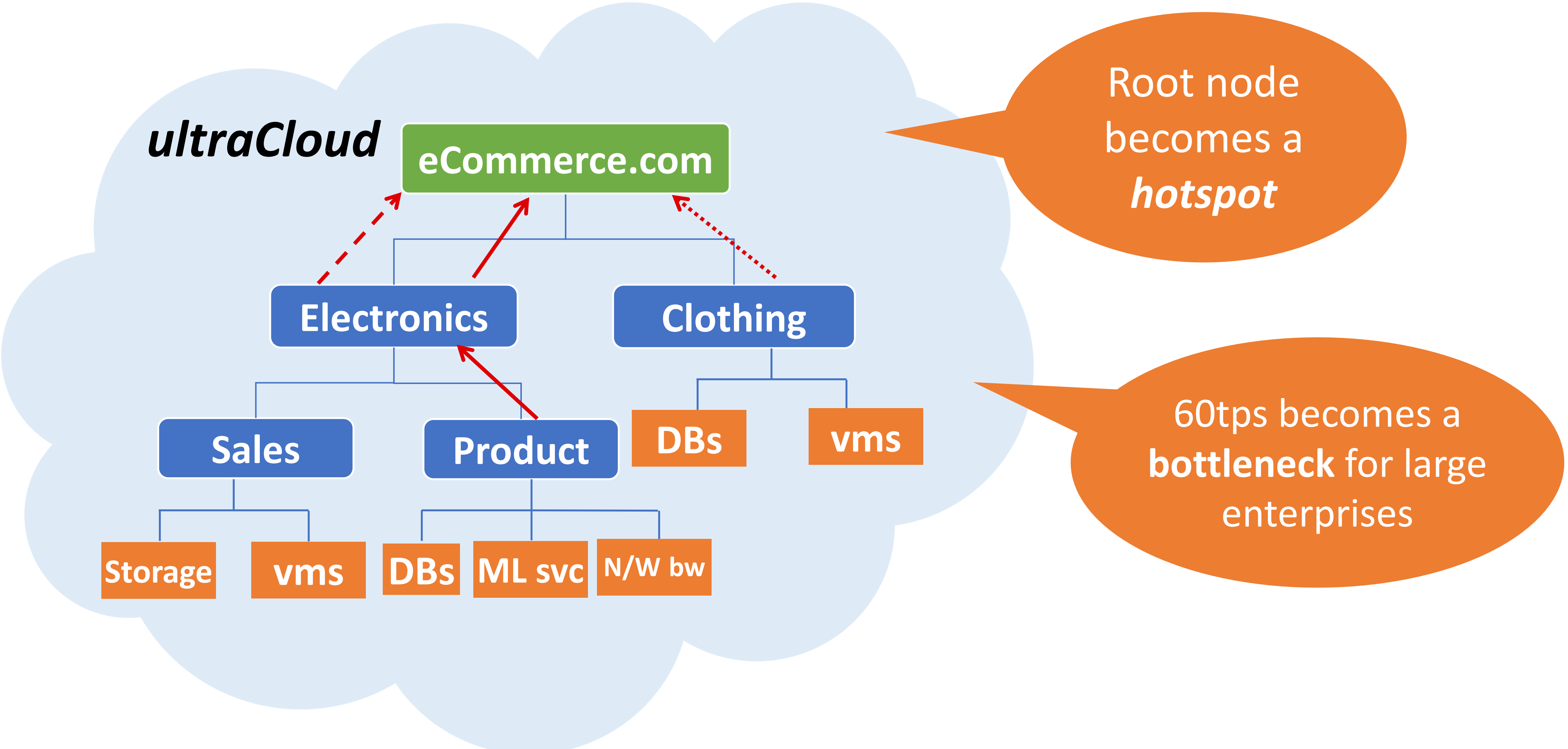
Consider an example: Resource management within a cloud provider



A max quota limit is set for each resource

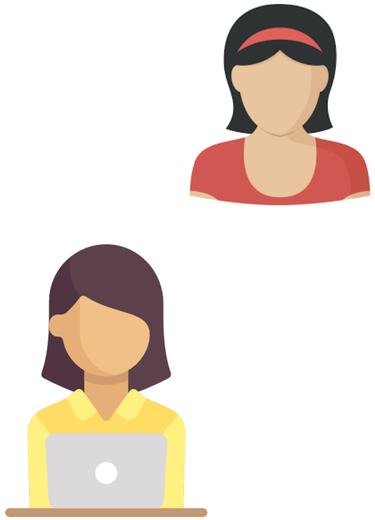
Individual teams *acquire* or *release* resources via read-write txns

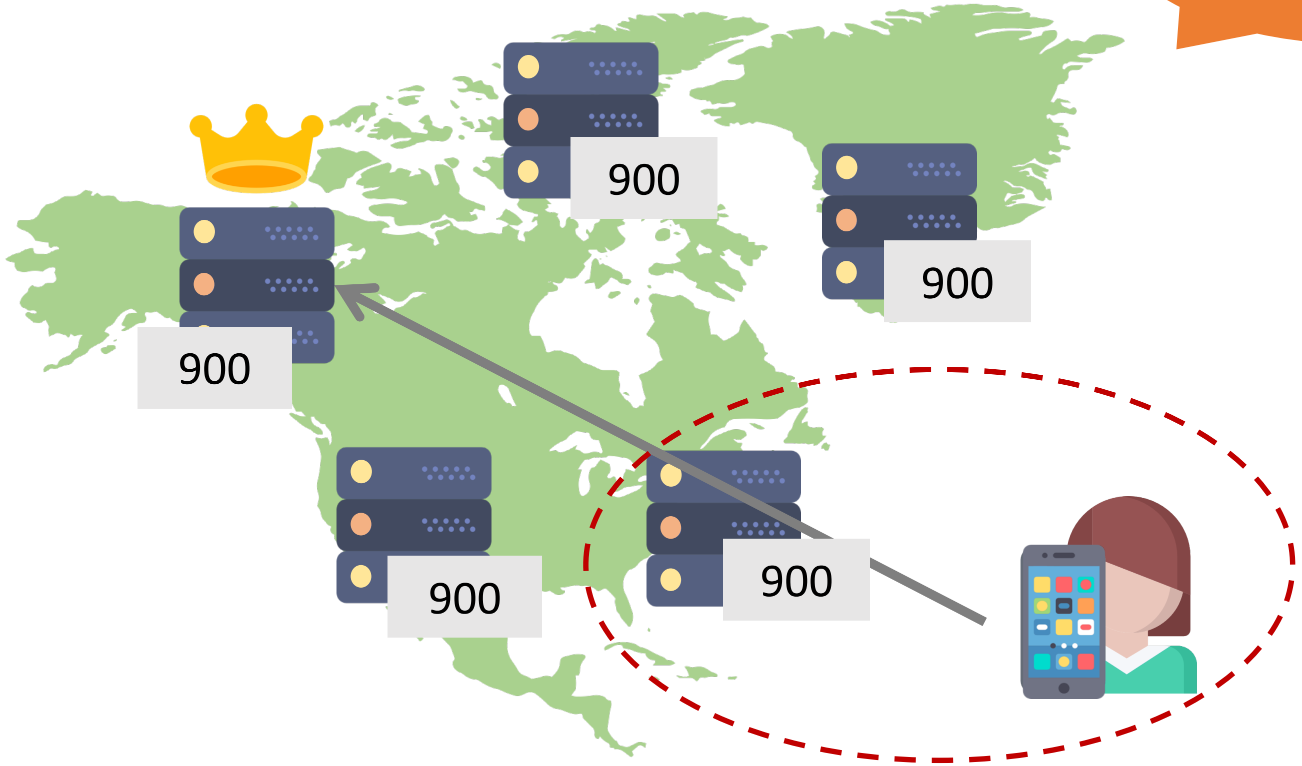
Consider an example: Resource management within a cloud provider



Issues with Spanner-like db design

1. **Sequential execution**
2. **Centralized, constant synchronization**
3. **Underutilized replicas**



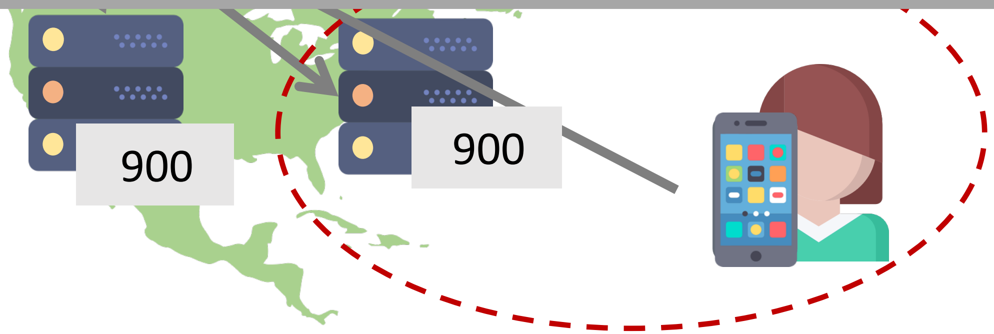


Issues with Spanner-like db design

1. **Sequential execution**
2. **Centralized, constant synchronization**
3. **Underutilized replicas**

- Manage aggregate data
- Update heavy workload

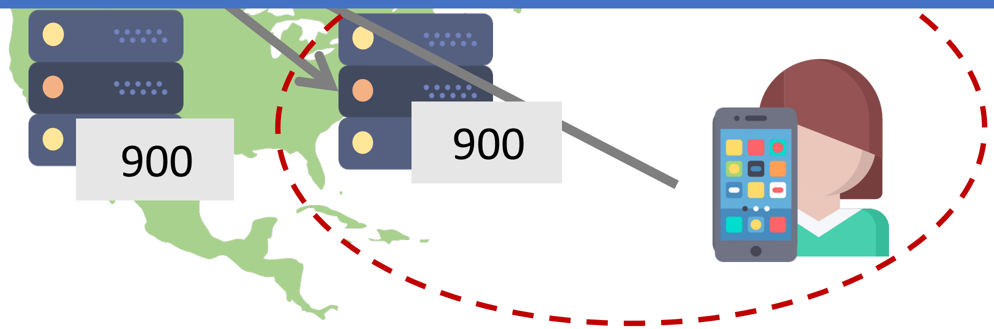
But low performance due to centralized, sequential execution



Issues with Spanner-like db design

1. **Sequential execution**
2. **Centralized, constant synchronization**
3. **Underutilized replicas**

Our research question:
Design an **alternate** system to manage
simple data types and provides ***high throughput*** for
update heavy workloads?



Looking back in the literature, we stumble upon many seminal works that answer our question..

O’Neil’s Escrow transactions [1]

Kumar and Stonebreaker [2]

Barbara and Garica-Molina’s Demarcation protocol [3]

Gustavo and El Abbadi [4]

[1] P. E. O’Neil. The escrow transactional method. ACM Transactions on Database Systems (TODS), 1986.
[2] A. Kumar and M. Stonebraker. Semantics based transaction management techniques for replicated data. ACM SIGMOD, 1988.
[3] D. Barbara and H. Garcia-Molina. The demarcation protocol: A technique for maintaining linear arithmetic constraints in distributed database systems. Springer, 1992
[4] G. Alonso and A. El Abbadi. Partitioned data objects in distributed databases. Distributed and Parallel Databases, 1995.

Looking back in the literature, we stumble upon many seminal works that answer our question..

O'Neil'

Barb

Dem

Partition the aggregate data and allow transactions to concurrently update different partitions

[1] P. E. O'Neil. The escrow transactional method. ACM Transactions on Database Systems (TODS), 1986.
[2] A. Kumar and M. Stonebraker. Semantics based transaction management techniques for replicated data. ACM SIGMOD, 1988.
[3] D. Barbara and H. Garcia-Molina. The demarcation protocol: A technique for maintaining linear arithmetic constraints in distributed database systems. Springer, 1992
[4] G. Alonso and A. El Abbadi. Partitioned data objects in distributed databases. Distributed and Parallel Databases, 1995.

Looking back in the literature, we stumble upon many seminal works that answer our question..

O’Neil

Barb

Dem

But proposed for radically different environments:

- sites are not geo-distributed
- networks are assumed reliable
- results are only simulations

[1] P. E. O’Neil. The escrow transactional method. ACM Transactions on Database Systems (TODS), 1986.

[2] A. Kumar and M. Stonebraker. Semantics based transaction management techniques for replicated data. ACM SIGMOD, 1988.

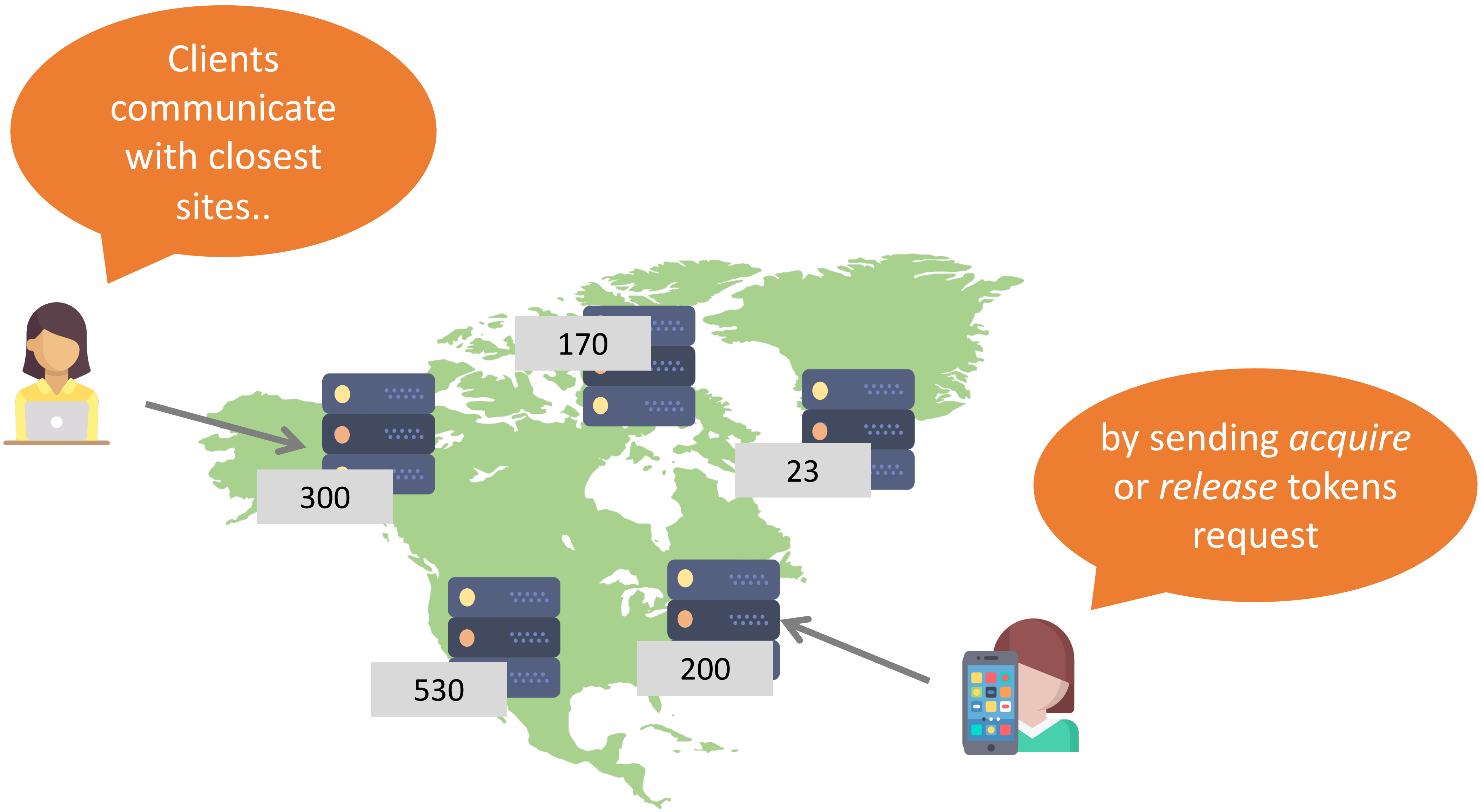
[3] D. Barbara and H. Garcia-Molina. The demarcation protocol: A technique for maintaining linear arithmetic constraints in distributed database systems. Springer, 1992

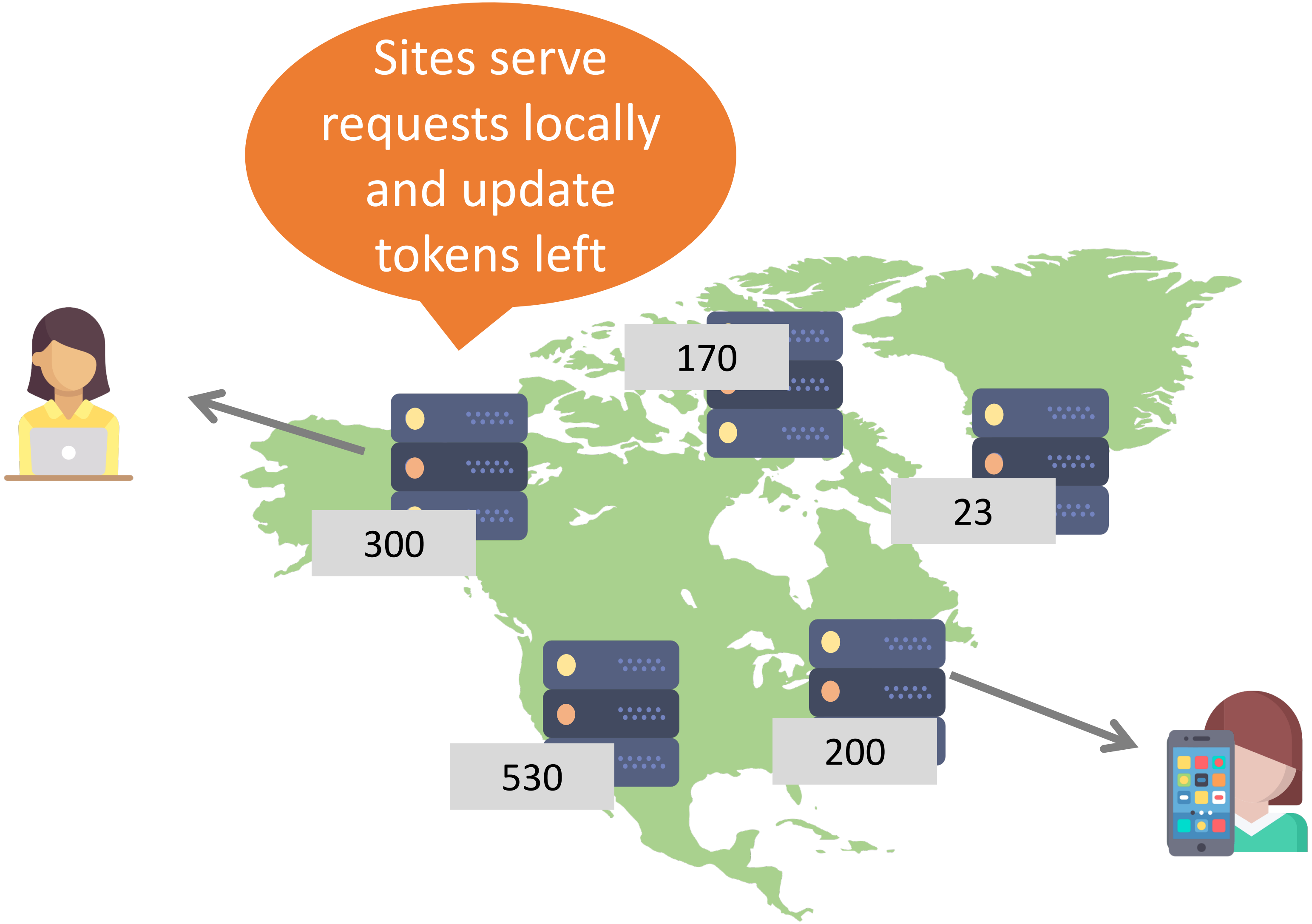
[4] G. Alonso and A. El Abbadi. Partitioned data objects in distributed databases. Distributed and Parallel Databases, 1995.

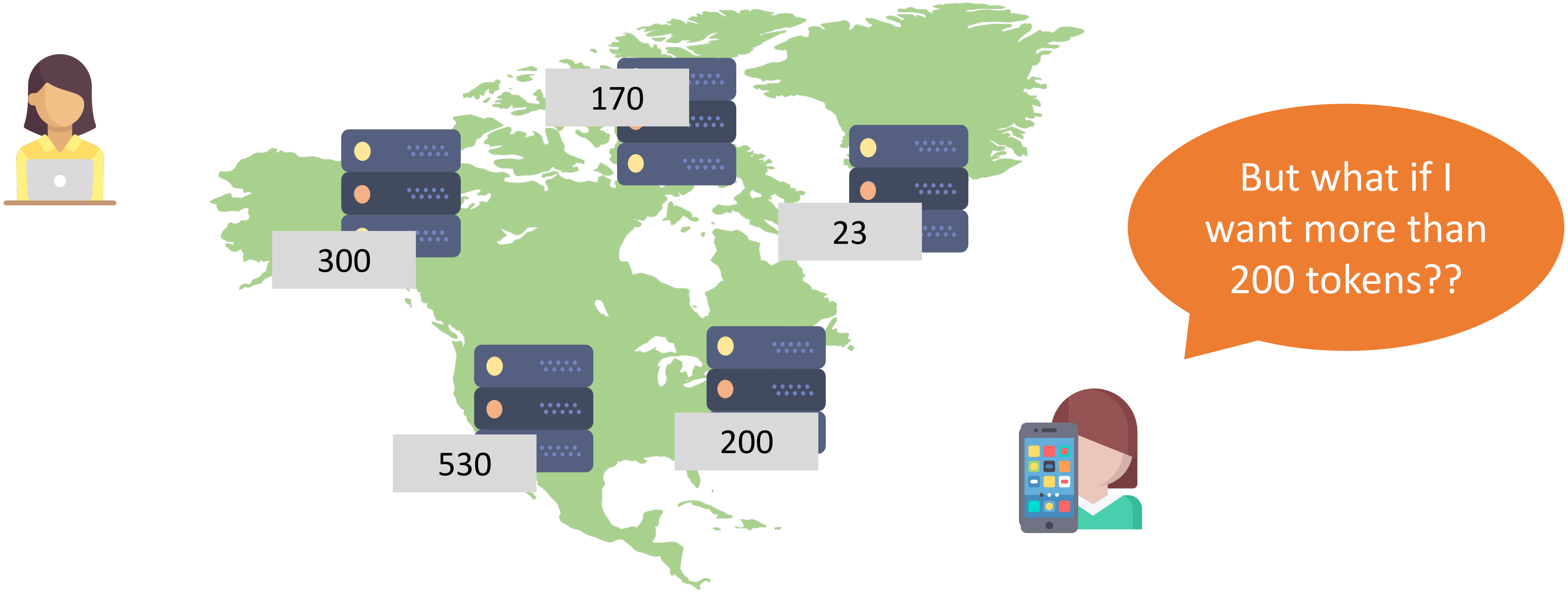
Samya brings the basic idea

– dis-aggregate the aggregate data to increase concurrency –

to the modern context of cloud and geo-distributed dbs


Clients communicate with closest sites..
170
300
23
by sending *acquire* or *release* tokens request
530
200


Sites serve requests locally and update tokens left
170
23
300
530
200


170
300
23
530
200
But what if I want more than 200 tokens??

Clients
commu
with

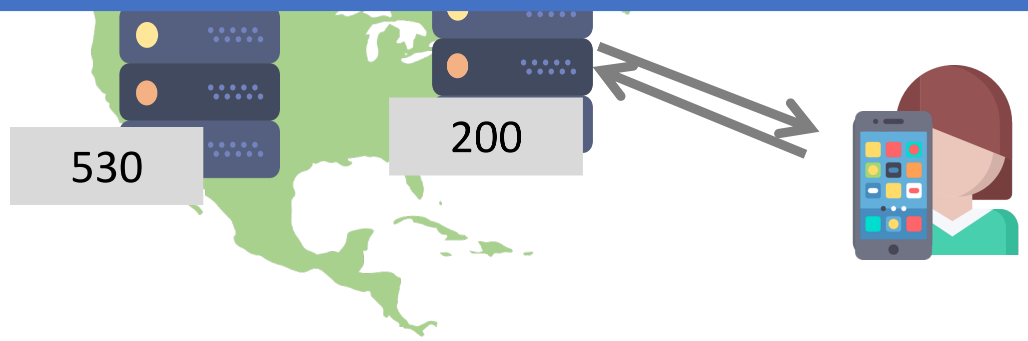
Sites serve
requests locally

Each site stores
disaggregated data

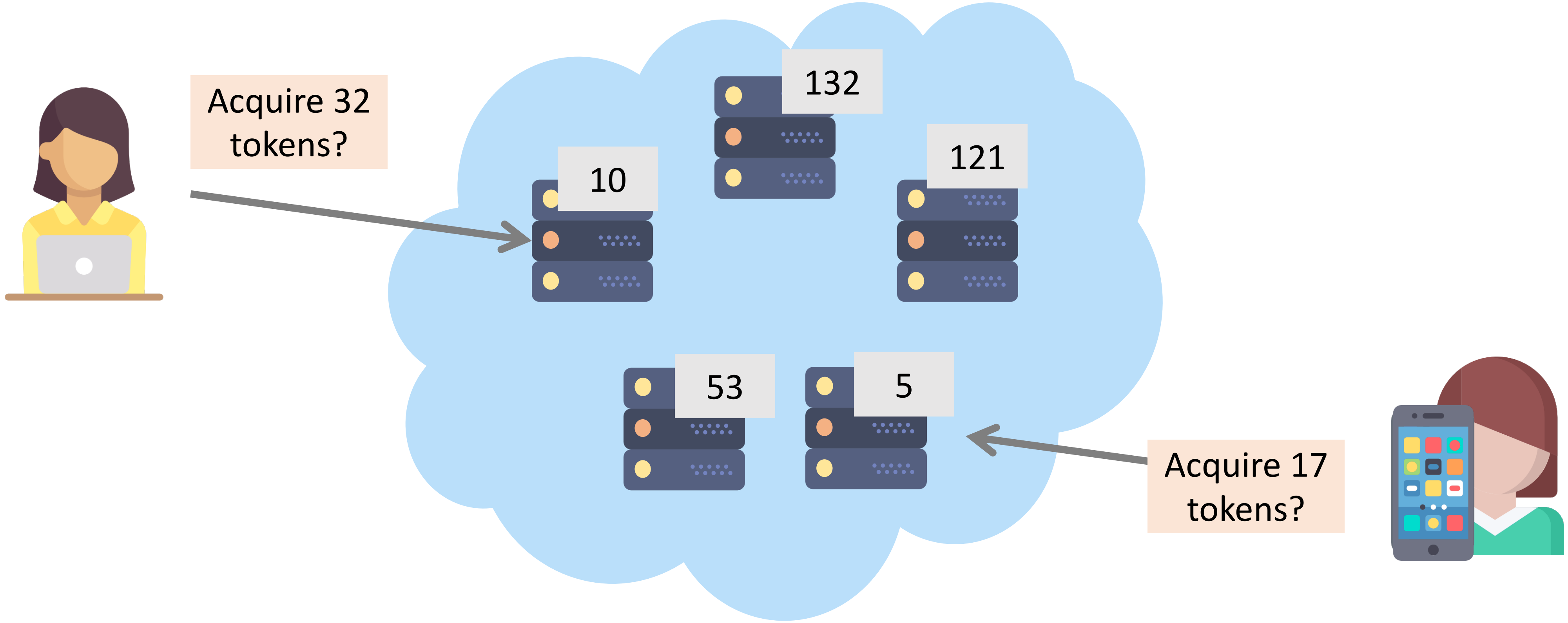
Avantan

a consensus protocol to agree on the global token availability and to ***redistribute*** tokens

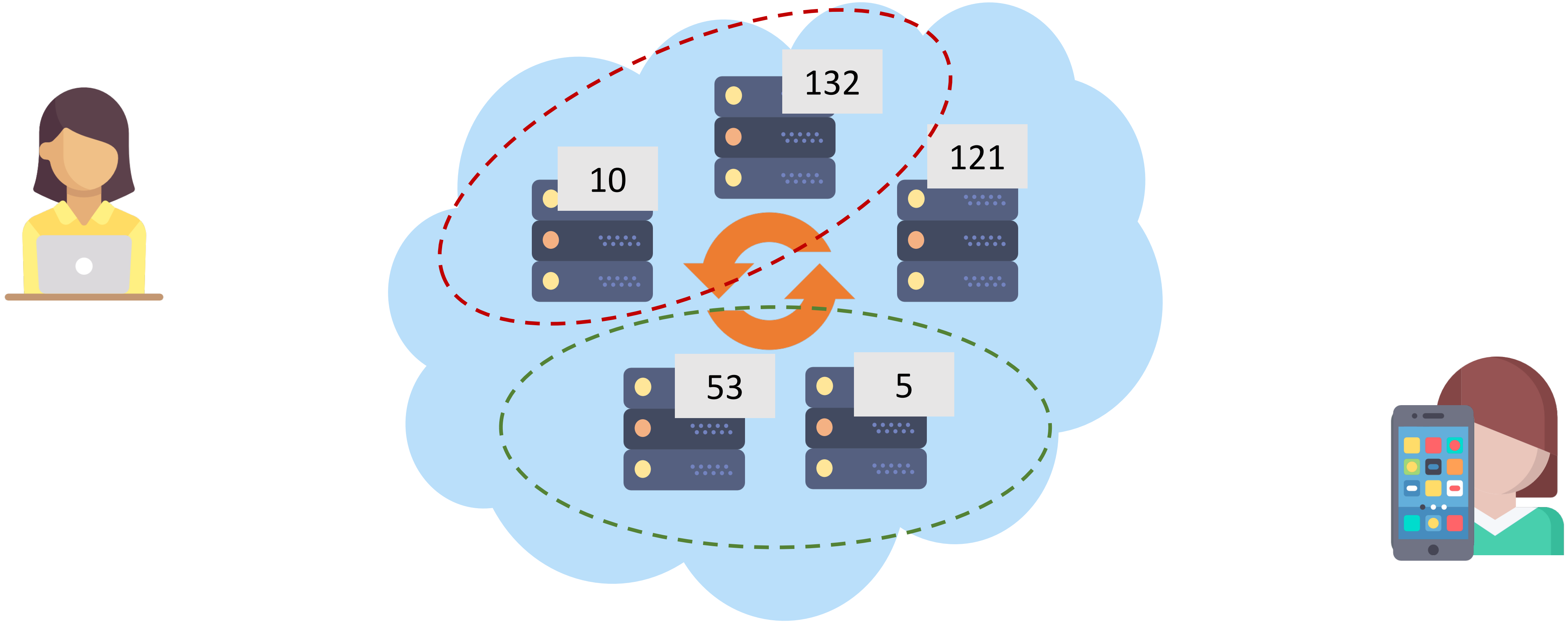
acquire
tokens
est



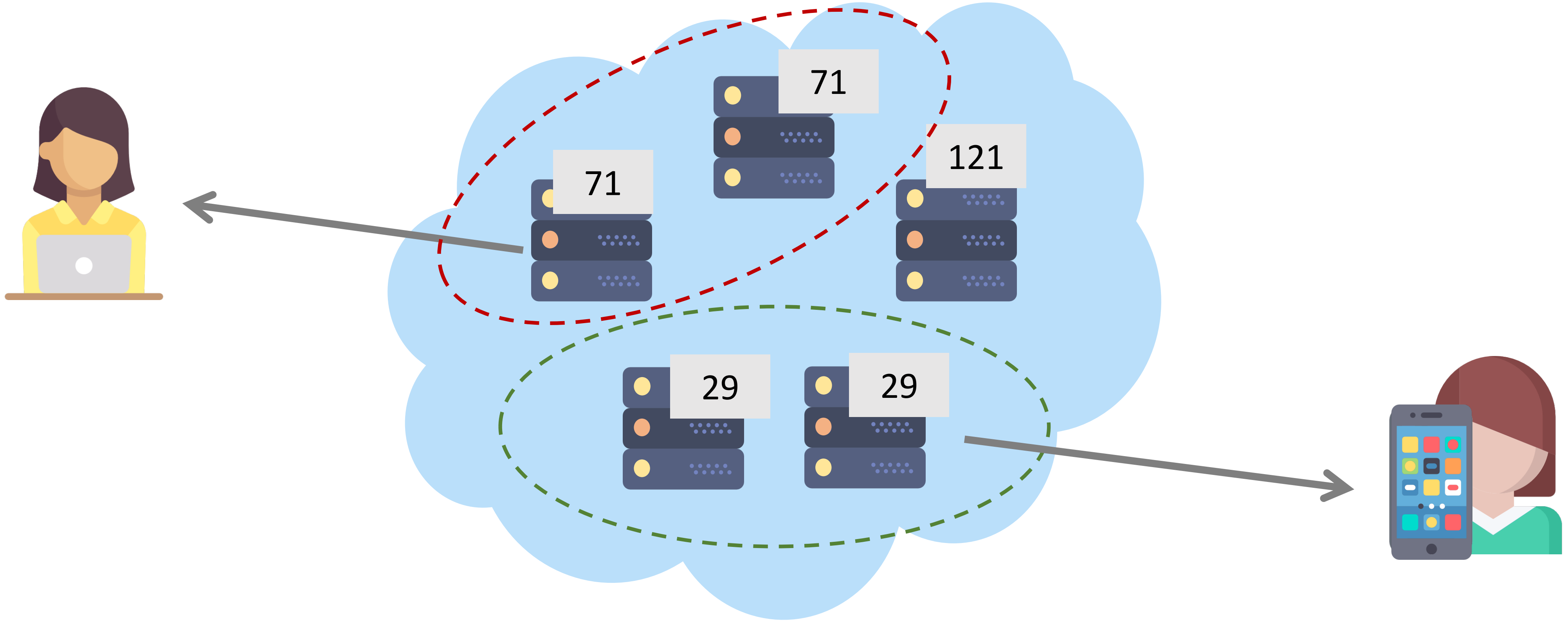
1. Avantan reaches agreement on ***available tokens*** – not on a client provided value
2. Avantan does ***not*** require a majority for consensus



1. Avantan reaches agreement on ***available tokens*** – not on a client provided value
2. Avantan does ***not*** require a majority for consensus



1. Avantan reaches agreement on ***available tokens*** – not on a client provided value
2. Avantan does ***not*** require a majority for consensus



But redistributing *after* a client sends request can cause lot of delay..

Demand predictions using
machine learning and deep learning to the rescue!!

Use analytical past resource demand data to predict future demands

When predicted demand increases, trigger ***proactive*** redistributions

Execute Avantan and borrow tokens from sites with decreasing demand

Evaluation setup

- **Servers/Clients**: GCP n1-standard VMs
- **Baselines**: Demarcation/Escrow, CockroachDB (Spanner-like db)
- **Dataset**: VM workload dataset by Microsoft Azure [1], inherently predictable workload
- **Prediction method**: Neural Networks (LSTMs)

Performance analysis of Samya

Samya commits **16x** to **18x** more transactions than CockroachDB

Although redistributions are expensive, redistributions increases Samya’s throughput by **14%**

Samya performs about **1.4x** better with predictions

If app. workload has **less** than 35% writes, Spanner-like DB performs better than Samya

Summary

- Samya: a data system for high-contention aggregate data
- Avantan is a novel consensus protocol used for token redistribution that does not require a majority
- Dis-aggregation and executing Avantan allows Samya to commit **16x** to **18x** more transactions than a Spanner-like database
- Redistributions and demand predictions significantly increases Samya's performance