Towards a Robust Edge-Native Storage System

Presenter: Nikhil Sreekumar

Authors: Nikhil Sreekumar, Abhishek Chandra, Jon Weissman

Nov 12th, 2020

SEC

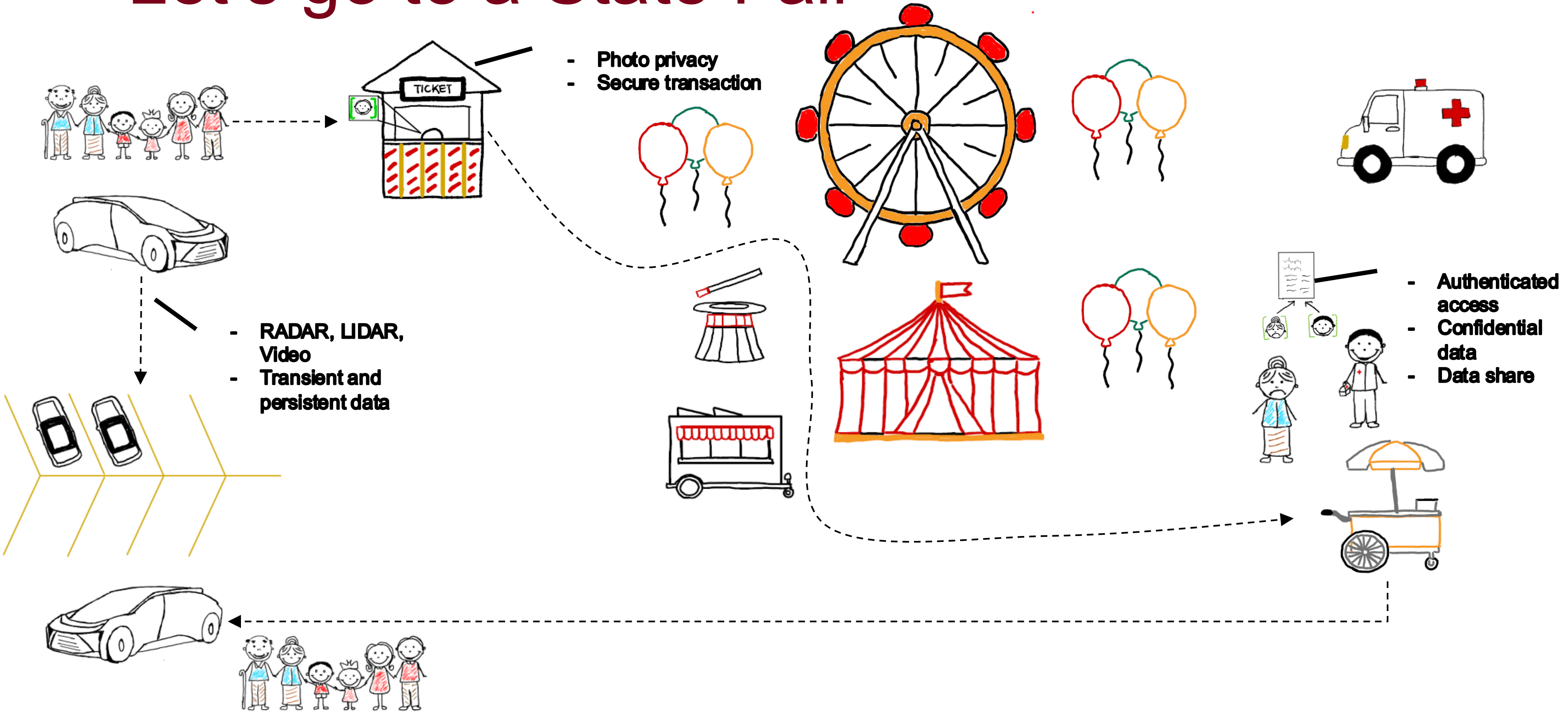
The Fifth ACM/IEEE Symposium on Edge Computing

Virtual, November 11-13, 2020

University of Minnesota

Driven to Discover®

Let’s go to a State Fair

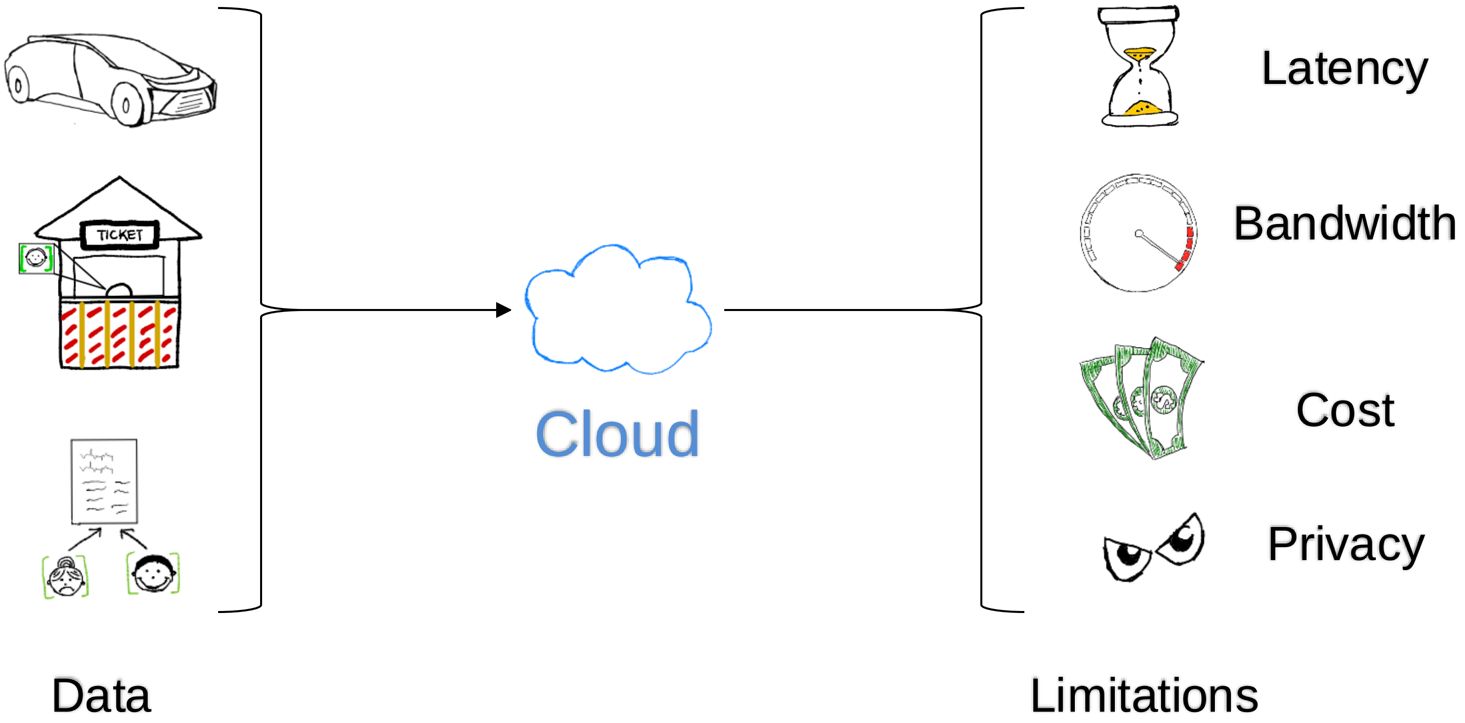


- Photo privacy
- Secure transaction

- RADAR, LIDAR, Video
- Transient and persistent data

- Authenticated access
- Confidential data
- Data share

Cloud solution



Cloud

Latency

Bandwidth

Cost

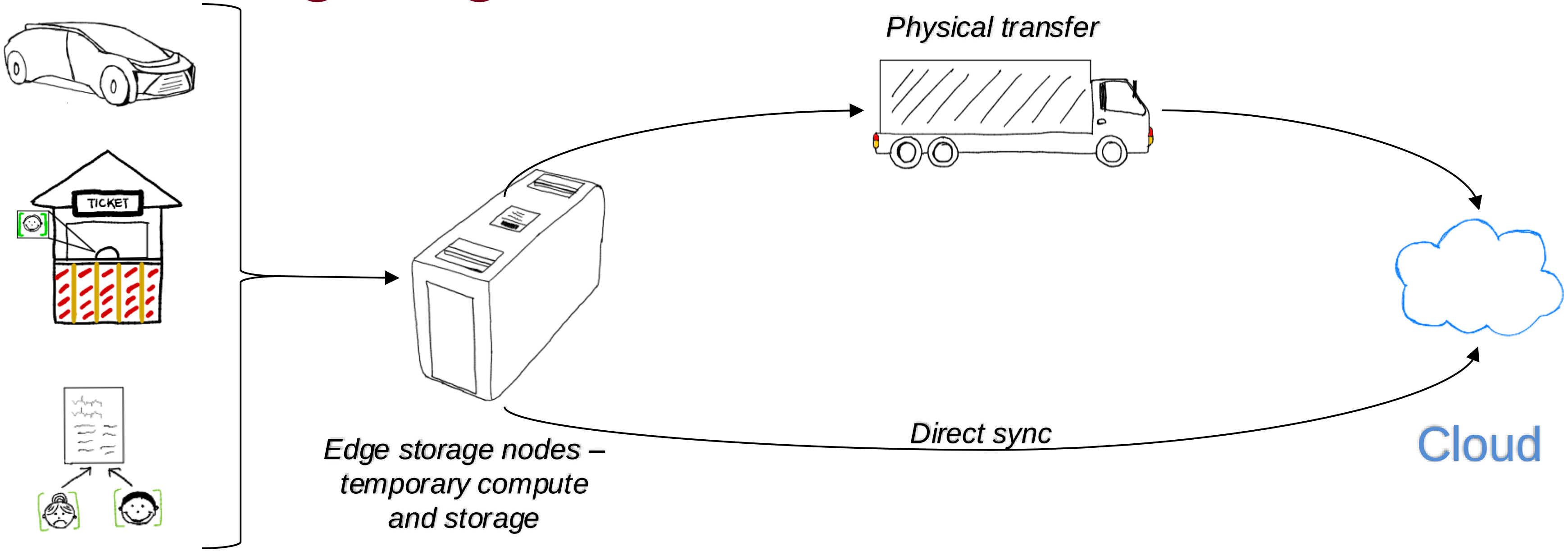
Privacy

Data

Limitations

This is where edge storage comes…

Existing Edge solutions



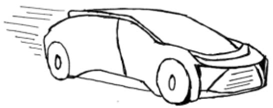
Physical transfer

Edge storage nodes – temporary compute and storage

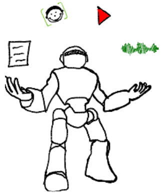
Direct sync

Cloud

Limitations

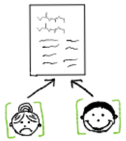


Mobility



Data management

Privacy of data



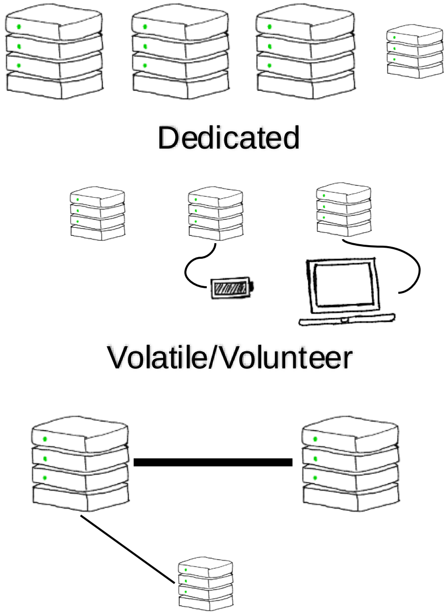
Low latency data sharing

Vision

An edge-native storage system that can operate anywhere with minimal infrastructure requirements by utilizing both pre-deployed and volatile/voluntary resources, catered to the needs of edge applications.

Challenges

Heterogeneity and churn

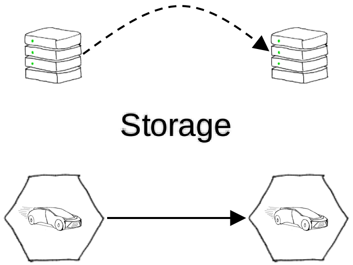


Dedicated

Volatile/Volunteer

Bandwidth variation

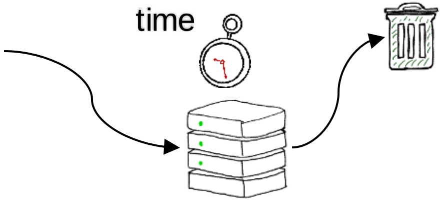
Data migration, replication, consistency



Storage

Mobile user/device

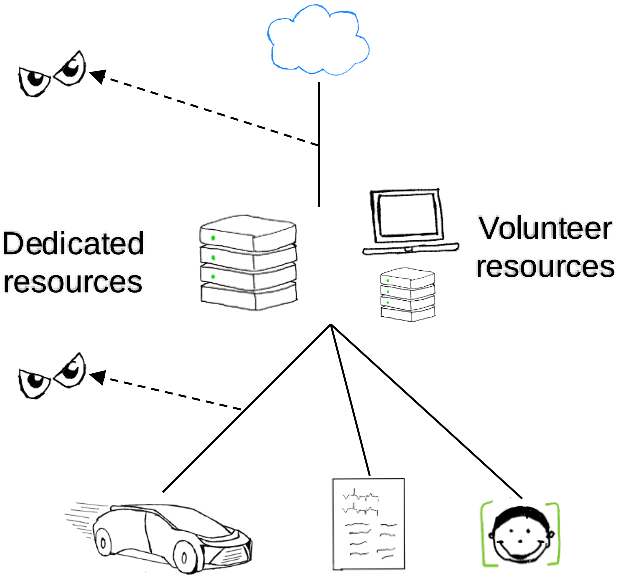
Data retention and discard



Retention time

Evict data

Privacy/security



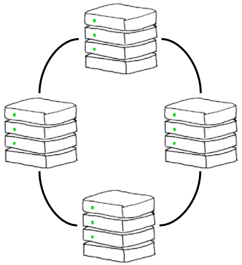
Dedicated resources

Volunteer resources

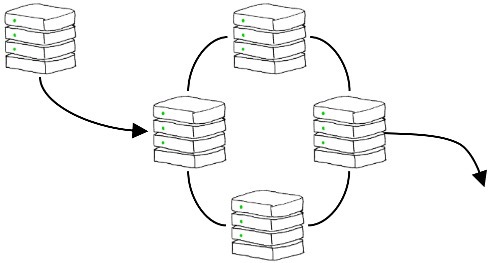
Can Cloud data management/storage solution be used at the Edge?

Decentralized, distributed, NoSQL database
High availability, performance and scalability

What makes Cassandra edge friendly?

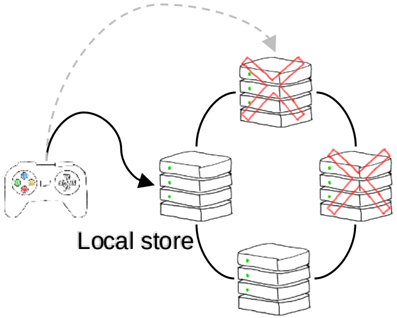


Decentralized & Distributed



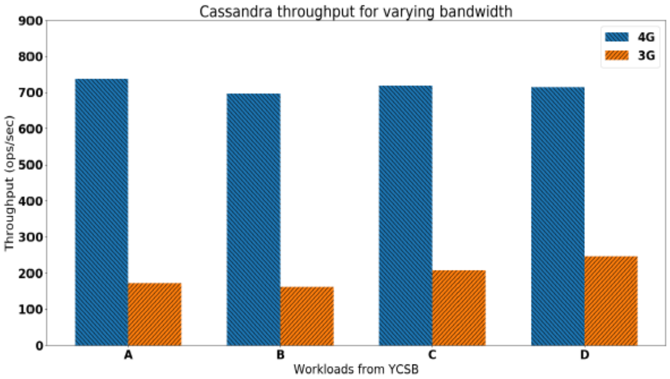
Scalable and Flexible

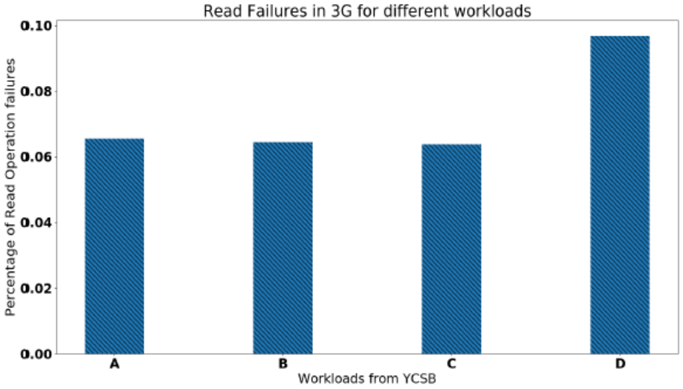
Fast writes



Hinted handoff/ Fault tolerance

Cassandra limitations - Constrained network bandwidth





YCSB Workload:

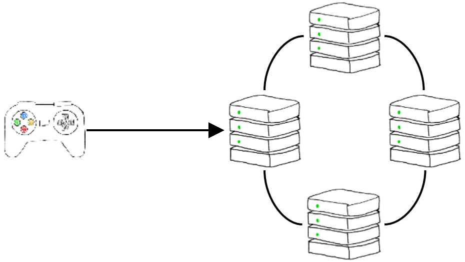
- ➢ Workloads A(50-50 read-write), B (95-5 read-write), C (100 read) and D (95-5 read-insert) - 10000 ops

Inference

- ➢ *Fails to perform in low bandwidth situations*

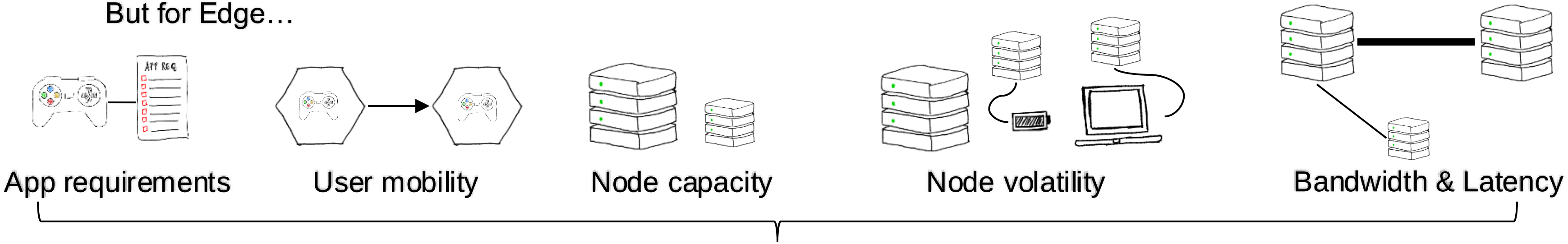
Cassandra limitations – Data placement and replication

Data placement and replication



Consistent hashing used to identify location of data storage

But for Edge…



App requirements

User mobility

Node capacity

Node volatility

Bandwidth & Latency

Decides data placement and replication strategy

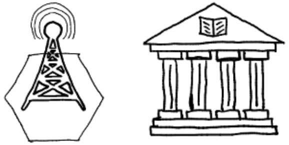
Design Principles

Dependent on

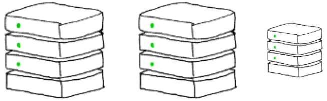
- Application requirements
- Existing cloud principles
- QoS
- User behavior

QoS-driven storage location/tier selection

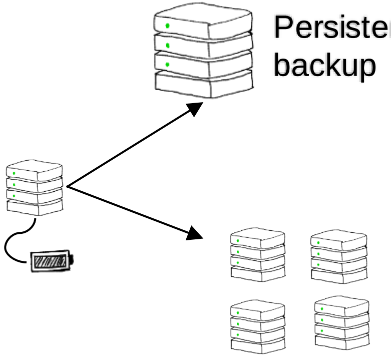
Storage install location



Dynamic selection of tiers



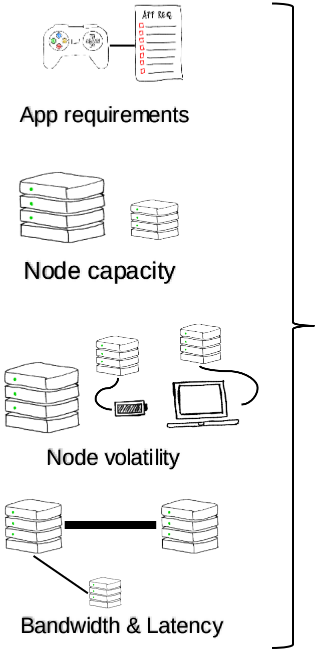
Volatile tier



Persistent store backup

Erasure coding

Context/mobility aware data placement



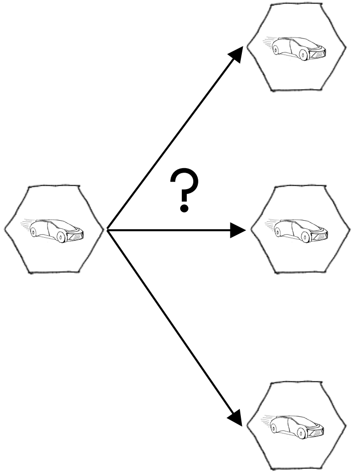
App requirements

Node capacity

Node volatility

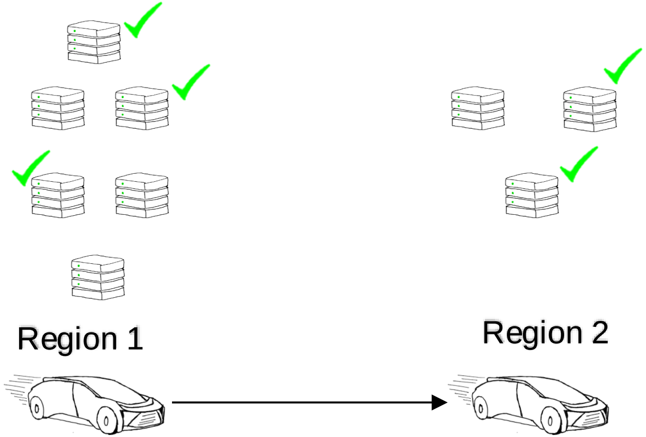
Bandwidth & Latency

Storage node selection

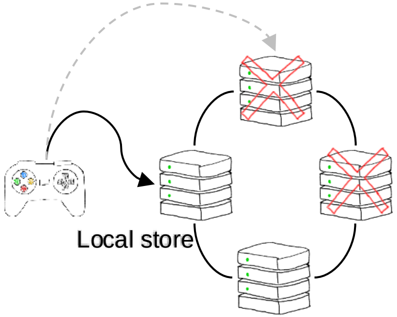


Prediction of next edge server region

Dynamic replication and hinted handoffs

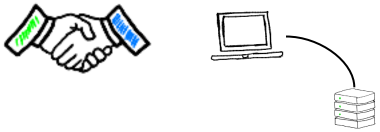


Change the replica count to adapt with resource limitation
Consistency policies changes dynamically

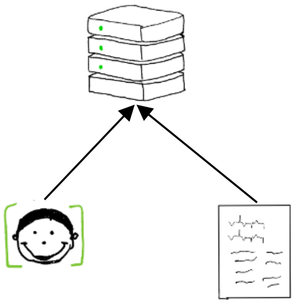


Hinted handoff for fault tolerance in high churn environment

Managed privacy

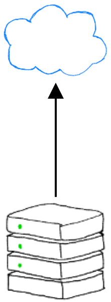


Trust management with volunteer resources



Use edge storage to store private data

Encryption, differential privacy

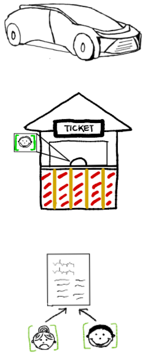


Denature data before sending to cloud

Filter private data

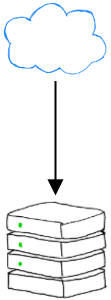
Obfuscation, secure aggregation in ML

Conclusion



App Scenario

Challenges



Cloud storage
tool at the Edge?

Design Principles

We believe a future edge storage system must be decentralized, QoS-driven, user/mobility aware and dynamic

Nikhil Sreekumar
sreek012@umn.edu

Abhishek Chandra
chandra@umn.edu

Jon Weissman
jon@cs.umn.edu

NSF Grants: CNS-1908566 and CNS-1619254

Distributed Computing Systems Group
University of Minnesota

UNIVERSITY OF MINNESOTA

Driven to Discover®

Crookston Duluth Morris Rochester Twin Cities