Microsoft

Research Faculty Summit 2018

Systems | Fueling future disruptions

Microsoft

Elevating the Edge to be a Peer of the Cloud

Kishore Ramachandran

Embedded Pervasive Lab, Georgia Tech

Acknowledgements

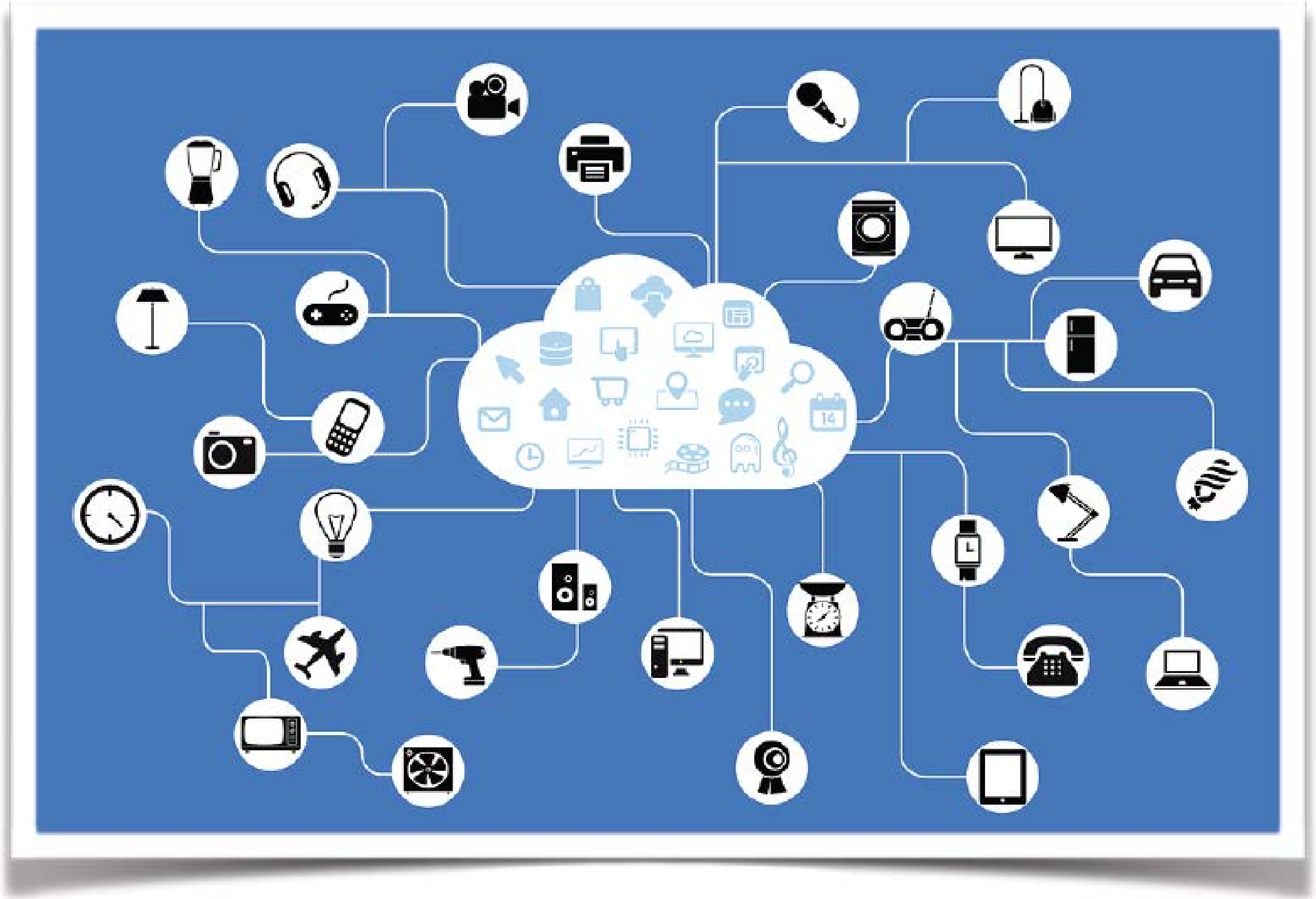
Enrique Saurez

Harshit Gupta

Zhuangdi Xu

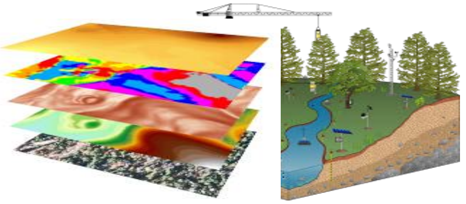
Adam Hall

IoT boom: Sensor-rich environment



A Broad Set of IoT Applications

Predictive maintenance

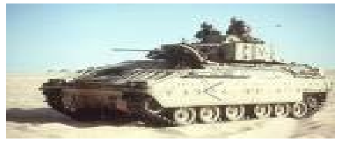


Enable New Knowledge

Energy Saving (I2E)

Transportation and Connected Vehicles

Intelligent Buildings

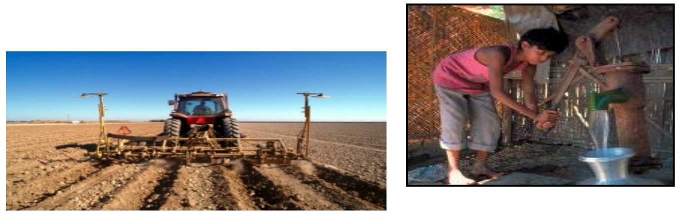


Defense

Industrial Automation

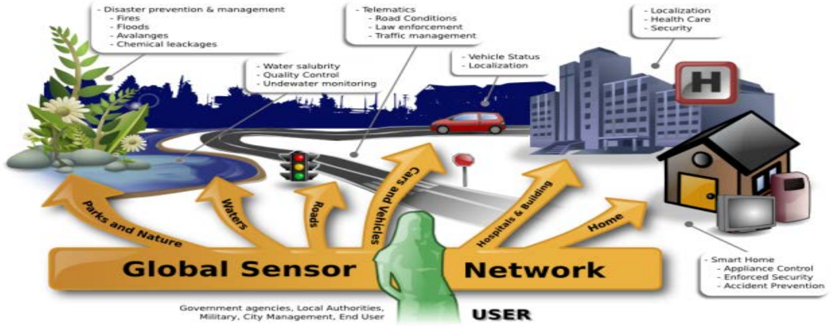
Enhance Safety & Security

Healthcare



Agriculture

Smart Grid



Smart Home

Thanks to CISCO for this slide

Future Internet Applications on IoT

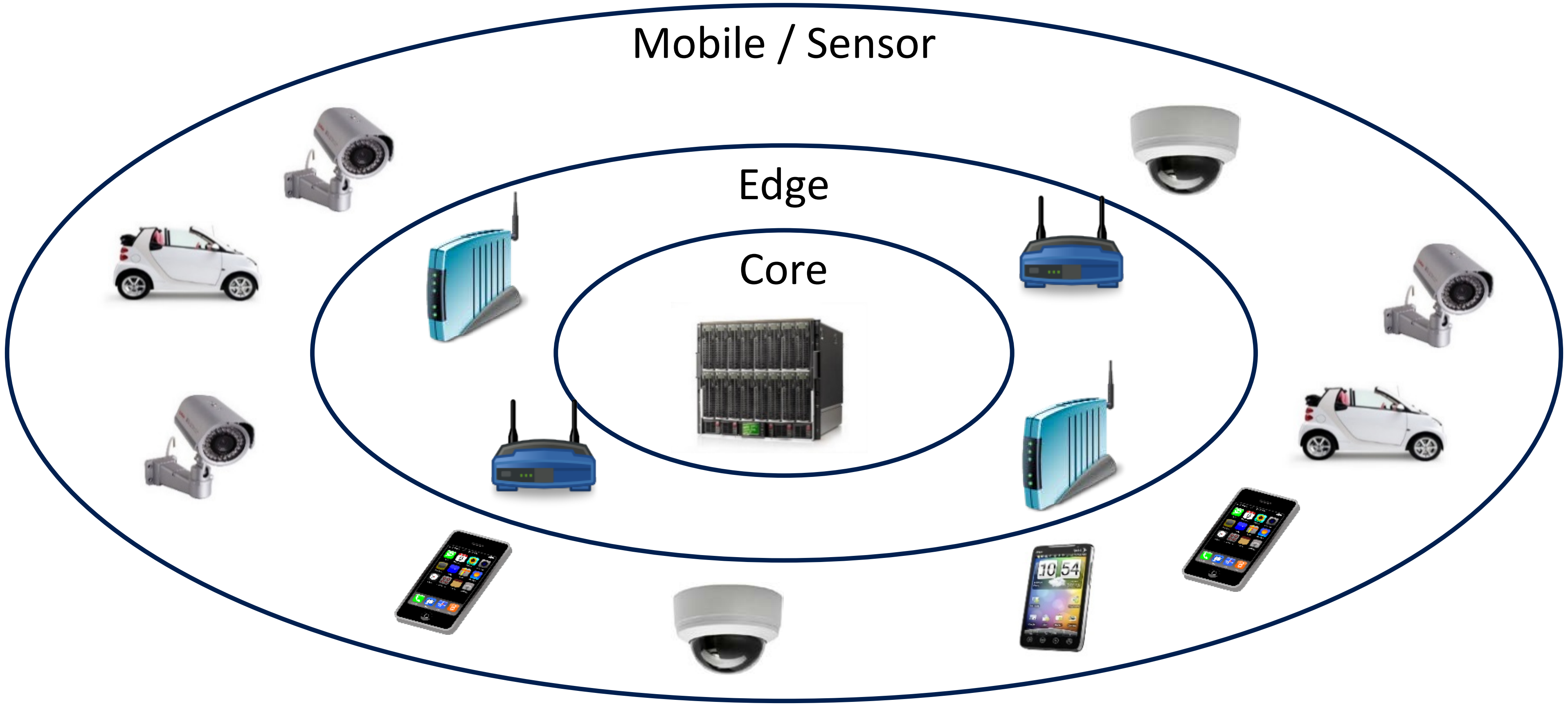
- Sense -> Process -> Actuate
- Common Characteristics
  - Dealing with real-world data streams
  - Real-time interaction among mobile devices
  - Wide-area analytics
- Requirements
  - Dynamic scalability
  - Low-latency communication
  - Efficient in-network processing

Cloud Computing

- Good for web apps at human perception speeds
  - Throughput oriented web apps with human in the loop
- Not good for many latency-sensitive IoT apps at computational perception speeds
  - sense -> process -> actuate
- Other considerations
  - Limited by backhaul bandwidth for transporting plethora of 24x7 sensor streams
  - Not all sensor streams meaningful
    => Quench the streams at the source
  - Privacy and regulatory requirements

Fog/Edge Computing

- Extending the cloud utility computing to the edge
- Provide utility computing using resources that are
  - Hierarchical
  - Geo-distributed



Fog/edge computing today

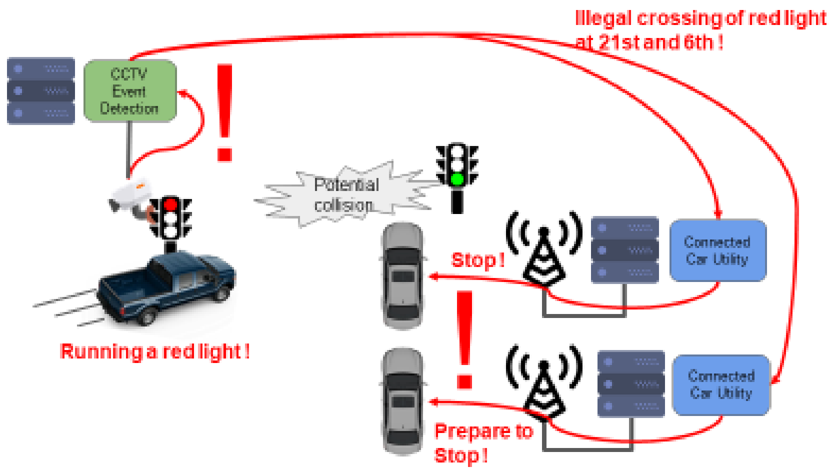
- Edge is slave of the Cloud
  - Platforms: IoT Azure Edge, CISCO Iox, Intel FRD, …
- Mobile apps beholden to the Cloud

Vision for the future

- Elevate Edge to be a peer of the Cloud
  - Prior art: Cloudlets (CMU+Microsoft), MAUI (Microsoft)
- In the limit
  - Make the Edge autonomous even if disconnected from the Cloud

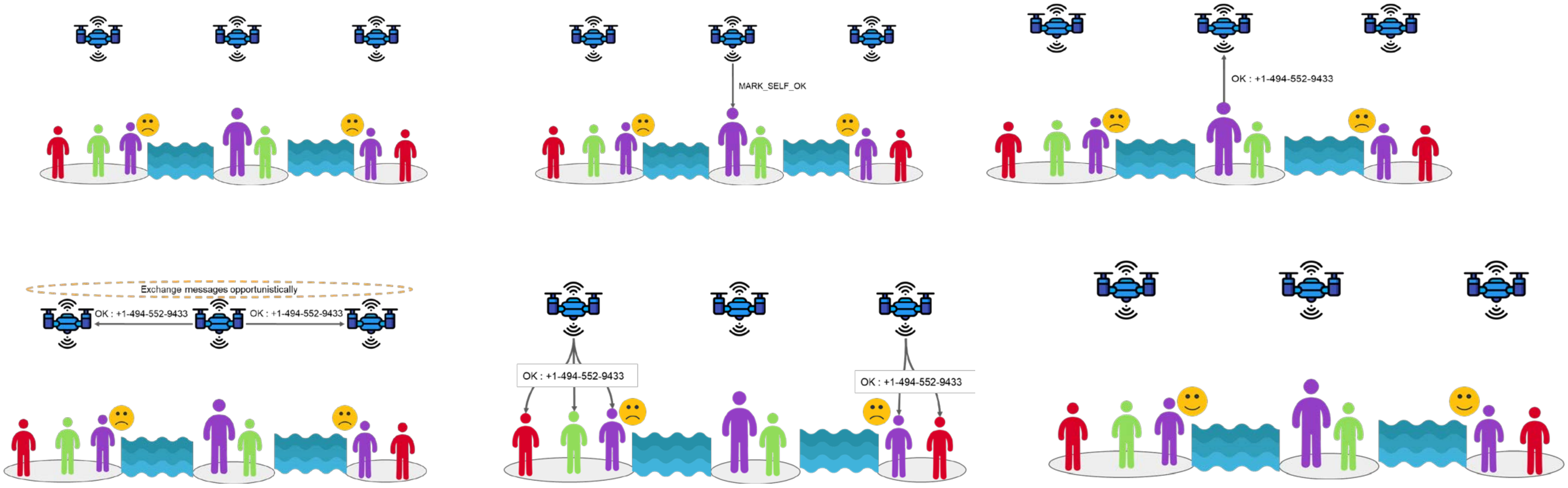
Why ?

- Interacting entities (e.g., connected vehicles) connected to different edge nodes
- Horizontal (p2p) interactions among edge nodes essential



Why ?

- Autonomy of edge (disaster recovery)



Challenges for making

- Need for powerful frameworks akin to the Cloud at the edge
  - Programming models, storage abstractions, pub/sub systems, …
- Geo-distributed data replication and consistency models
  - Heterogeneity of network resources
  - Resilience to coordinated power failures
- Rapid deployment of application components, multi-tenancy, and elasticity at the edge
  - Cognizant of limited computational, networking, and storage resources

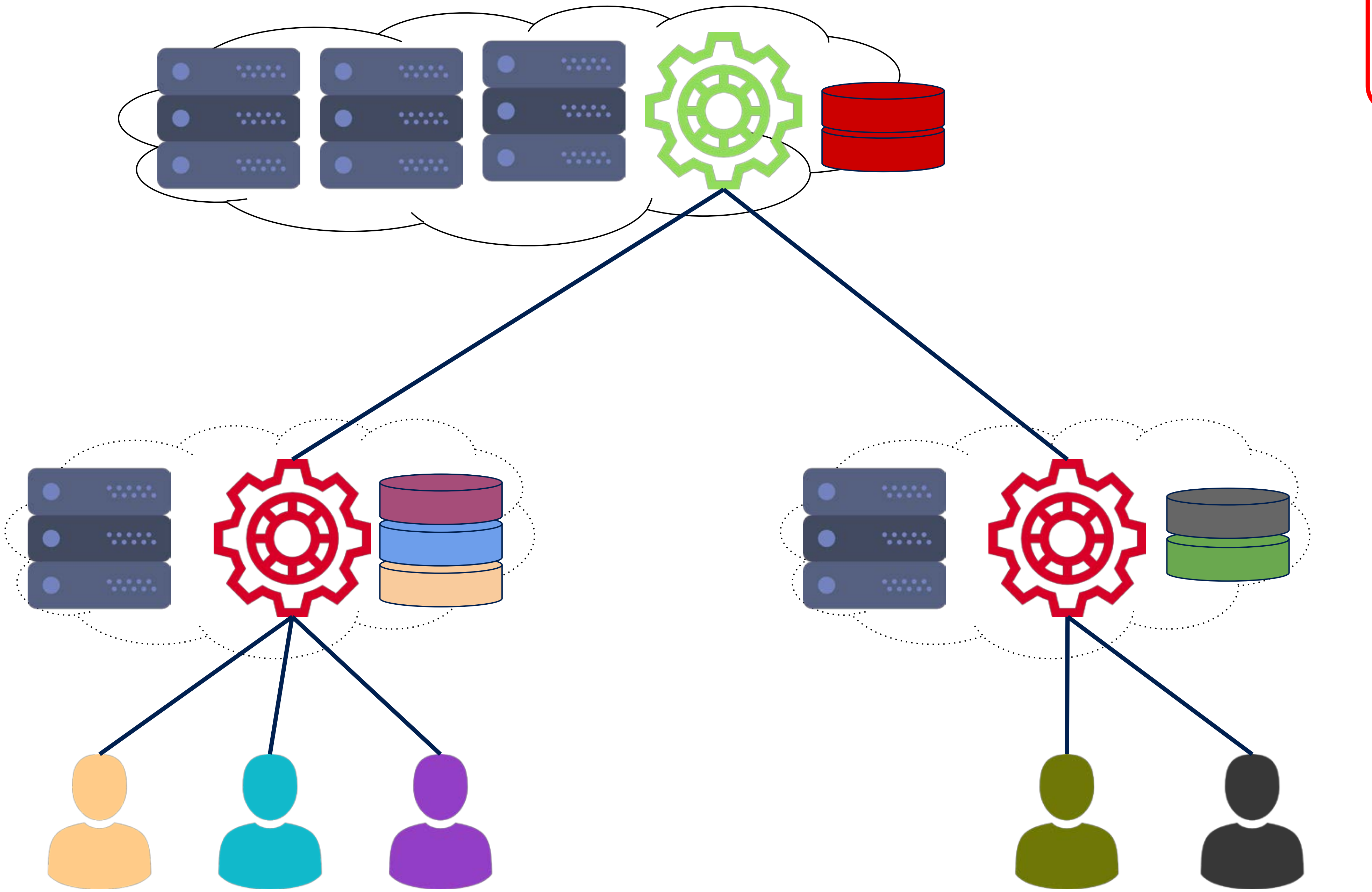
Thoughts on meeting these challenges

- Geo-distributed programming model for Edge/Cloud continuum
  - Foglets (ACM DEBS 2016)
- Geo-distributed data replication and resource management
  - FogStore (ACM DEBS 2018)
  - DataFog (HotEdge 2018)
- Applications using autonomous Edge
  - STTR: Space Time Trajectory Registration (ACM DEBS 2018)
  - Social Sensing *sans* Cloud (SocialSens 2017)

Geo-distributed programming model

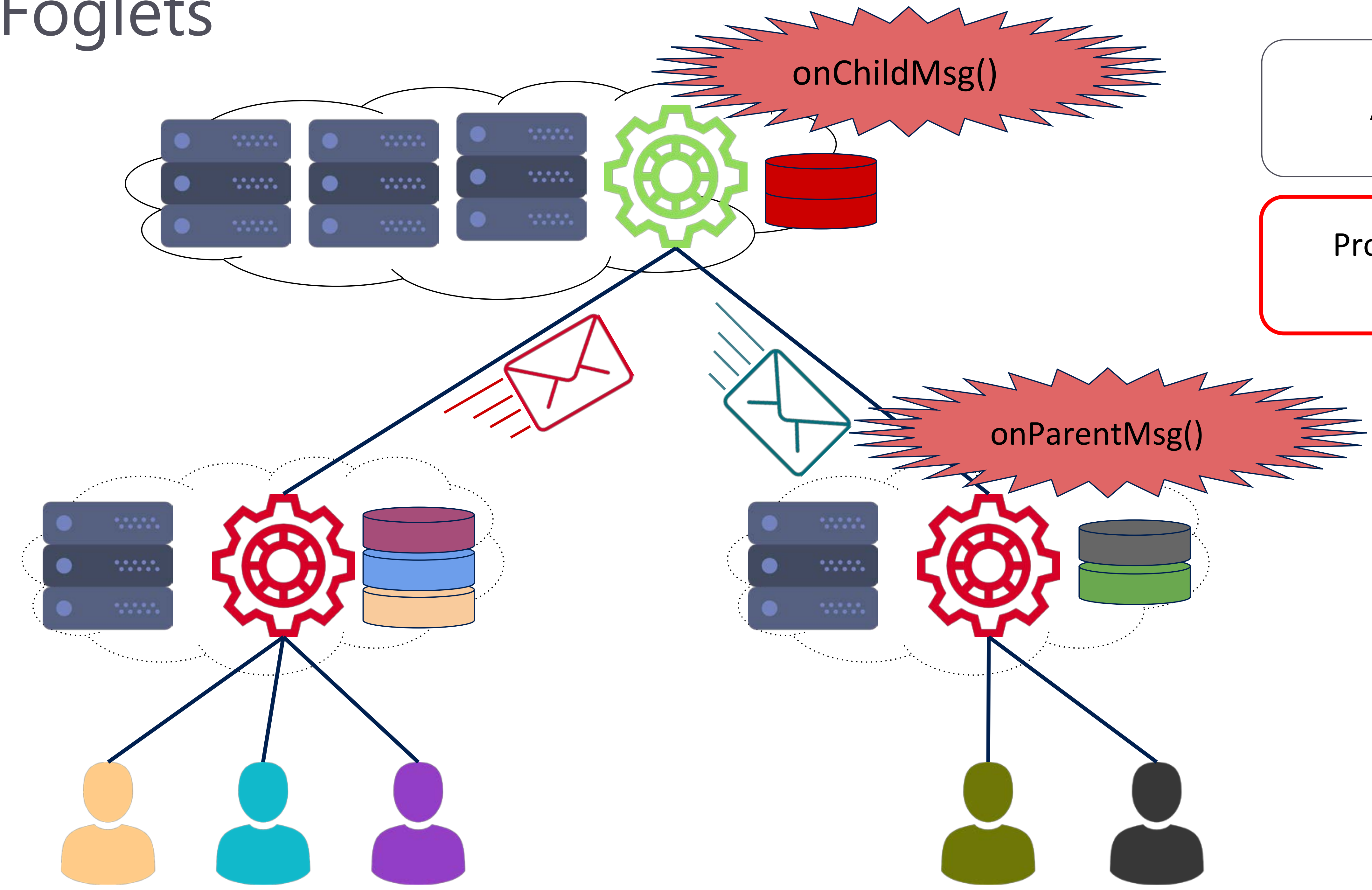
Foglets

App as dataflow graph



Geo-distributed programming model

Foglets

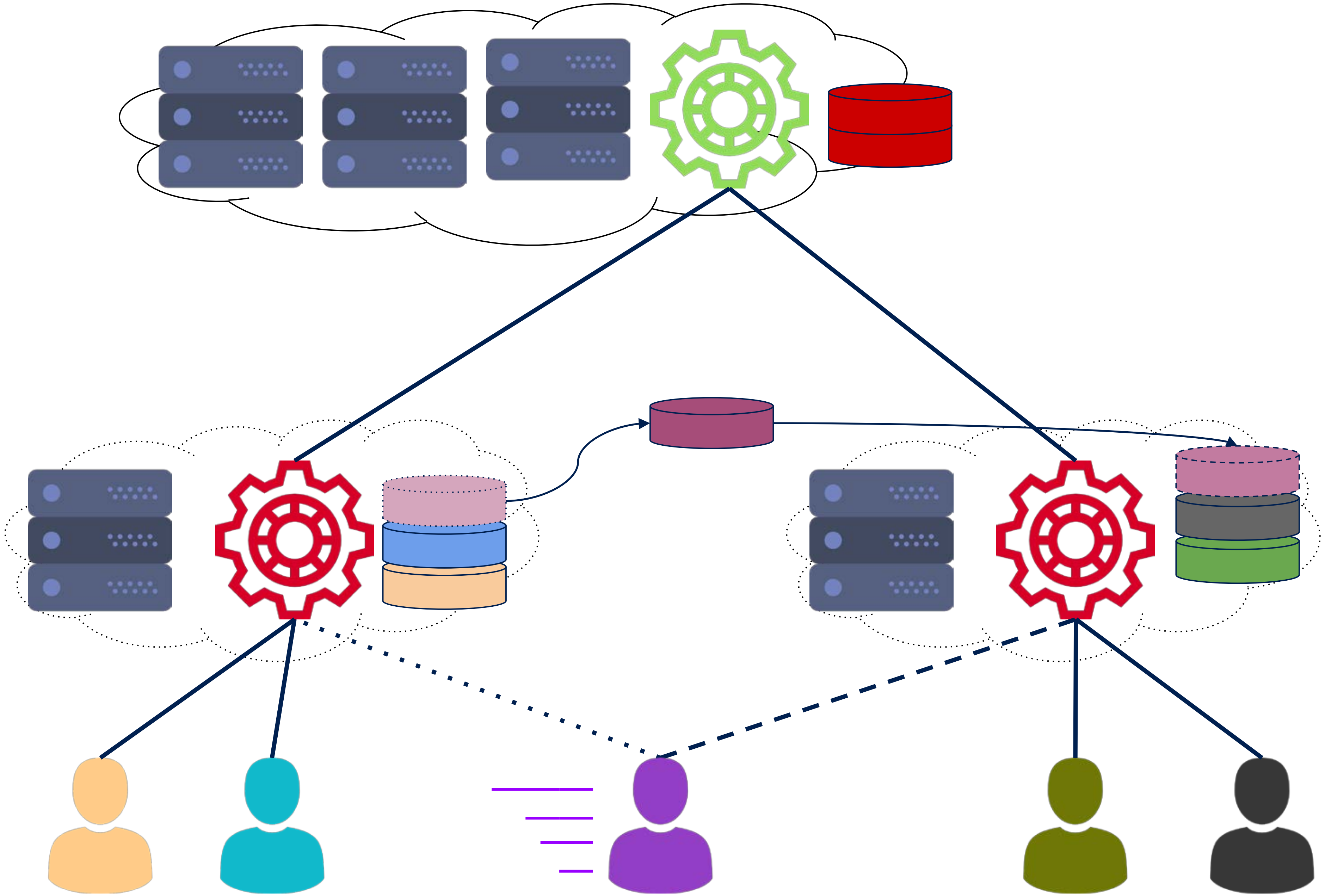


App as dataflow graph

Provides event handlers for communication

Geo-distributed programming model

Foglets



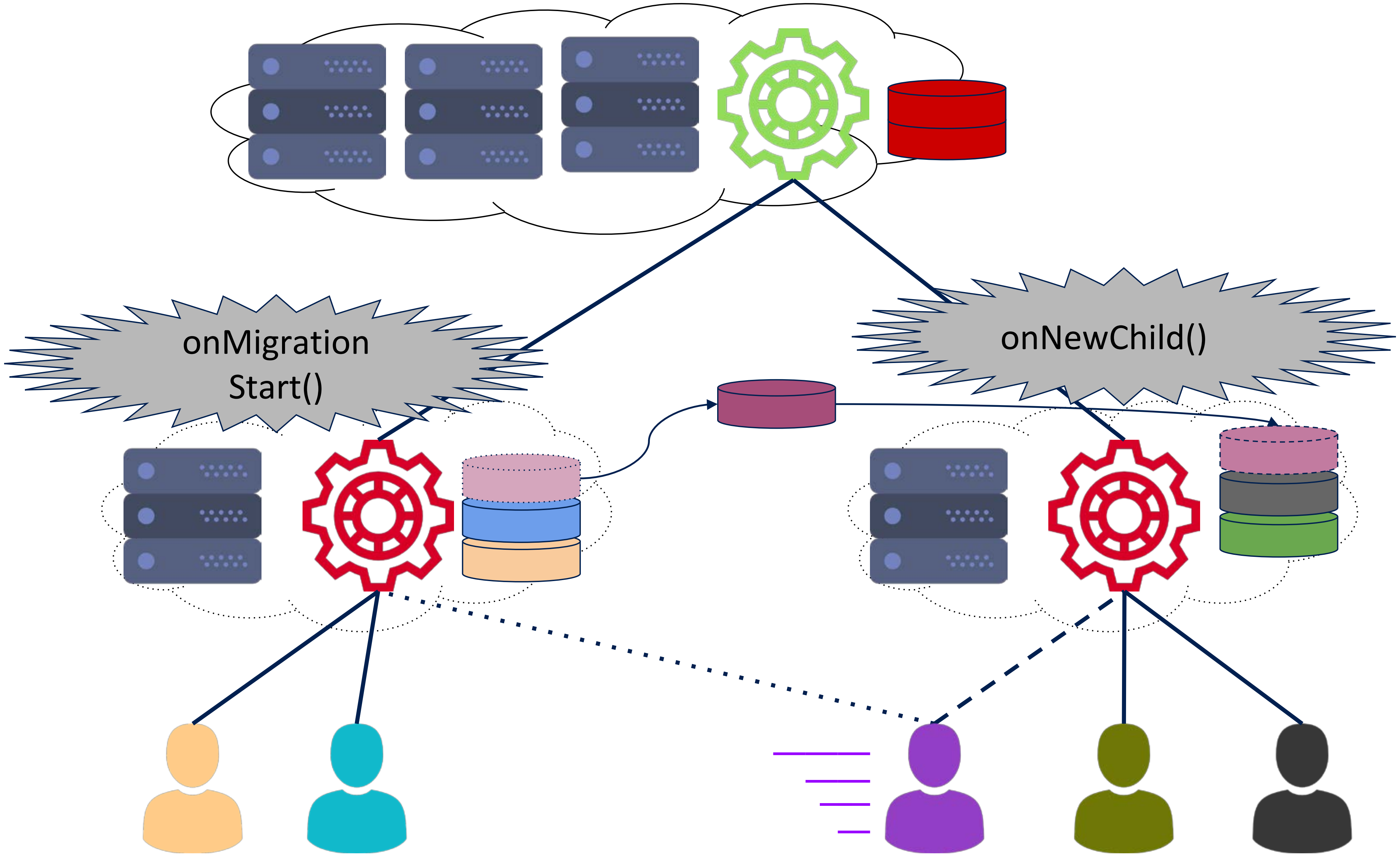
App as dataflow graph

Provides event handlers for communication

Transparent state migration

Geo-distributed programming model

Foglets



App as dataflow graph

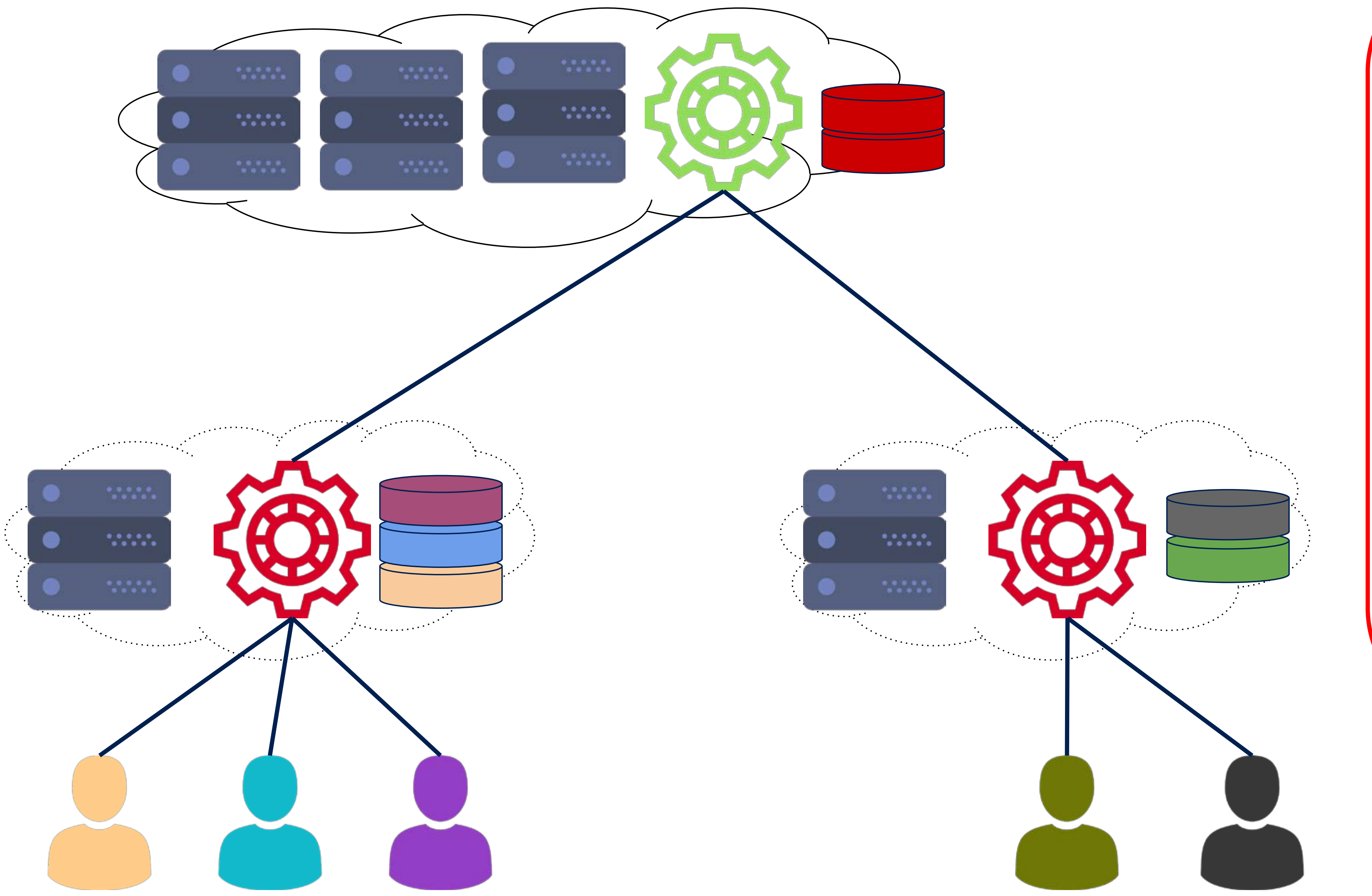
Provides event handlers for communication

Transparent state migration

Handlers for migration events

Geo-distributed programming model

Foglets

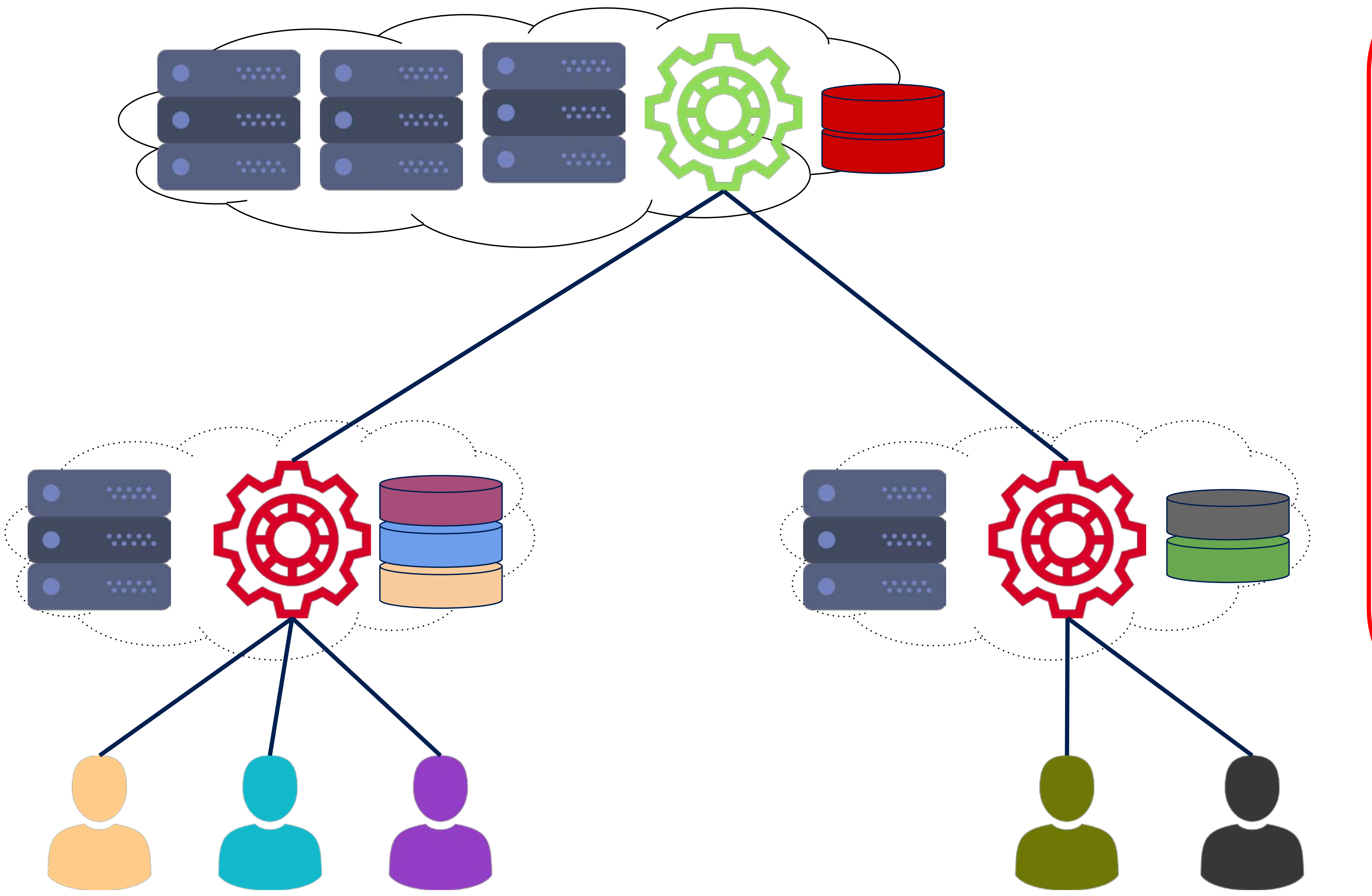


Summary:

- Sense->process->actuate app as a data flow graph with latency SLAs
- Auto discovery and placement of app components in Fog-Cloud continuum
- Migration of computation and state commensurate with mobility and/or resource constraints
- Spatio-temporal KV store for stashing state
- Multi-tenancy in the Fog nodes via virtualization

Geo-distributed programming model

Foglets



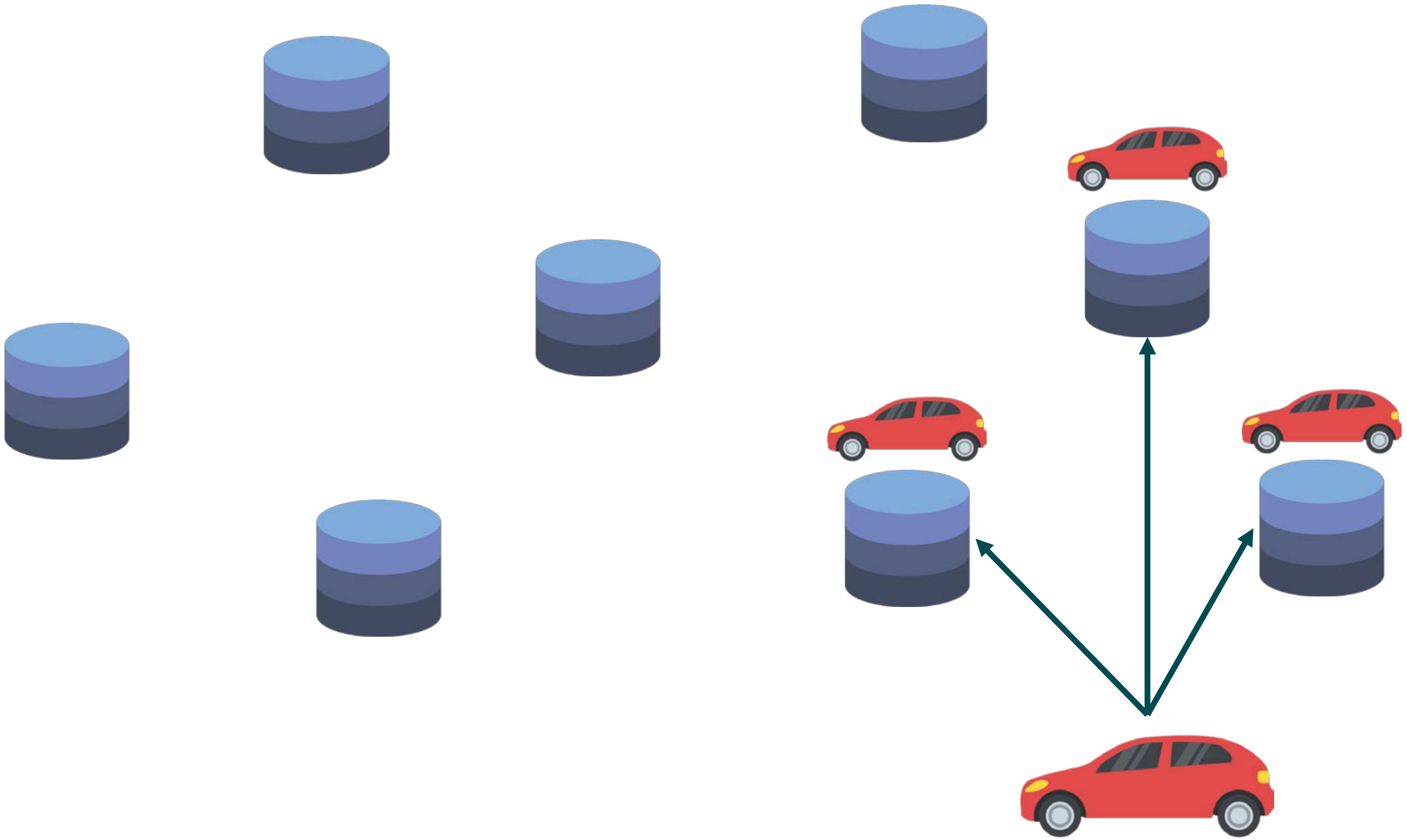
Summary:

- Sense->process->actuate app as a data flow graph with latency SLAs
- Auto discovery and placement of app components in Fog-Cloud continuum
- Migration of computation and state commensurate with mobility and/or resource constraints
- Spatio-temporal KV store for stashing state
- Multi-tenancy in the Fog nodes via virtualization

ACM DEBS 2016 for details

Replication: Consistency/Latency tradeoff

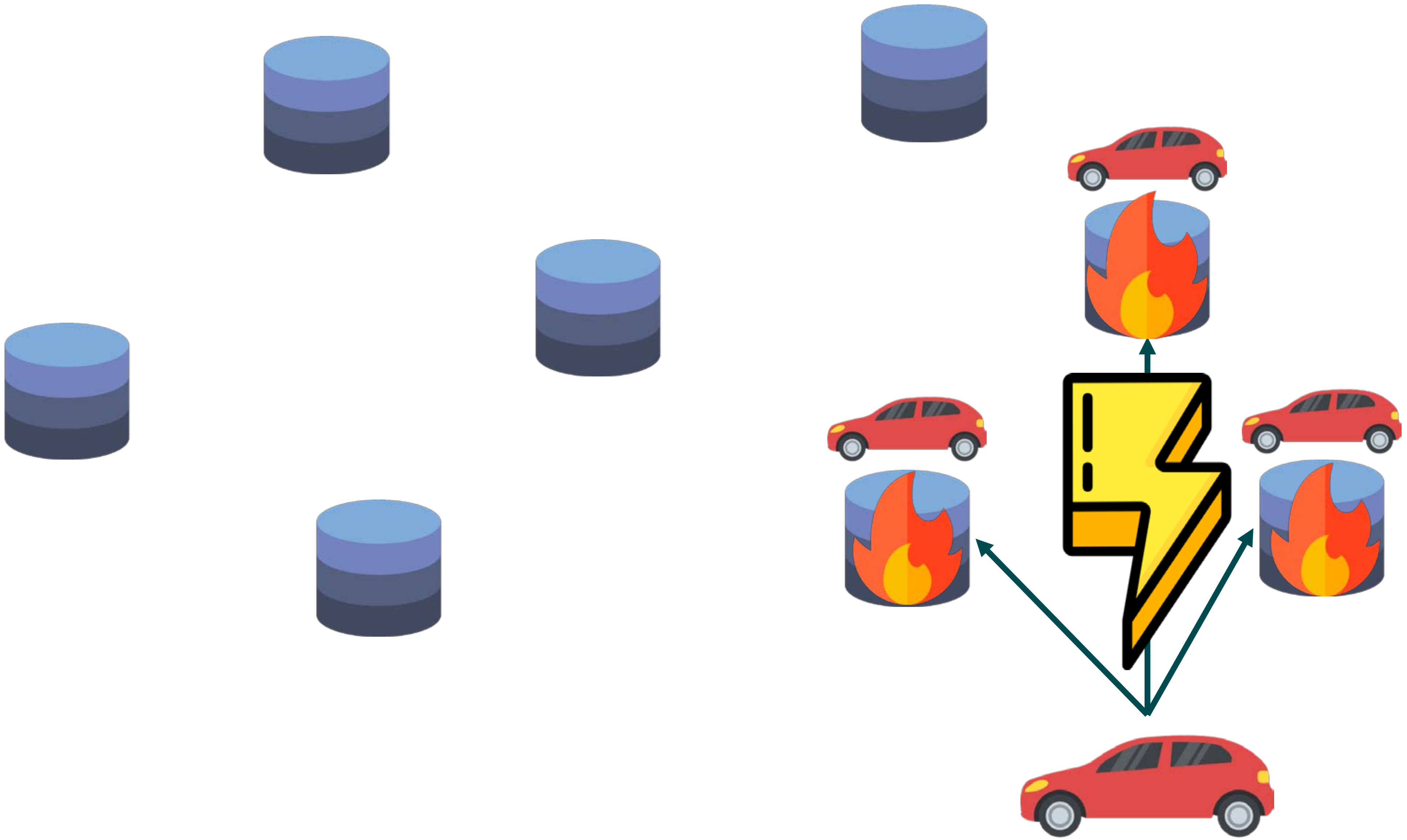
FogStore



Low-latency quorum read/update

Replication: Consistency/Latency tradeoff

FogStore

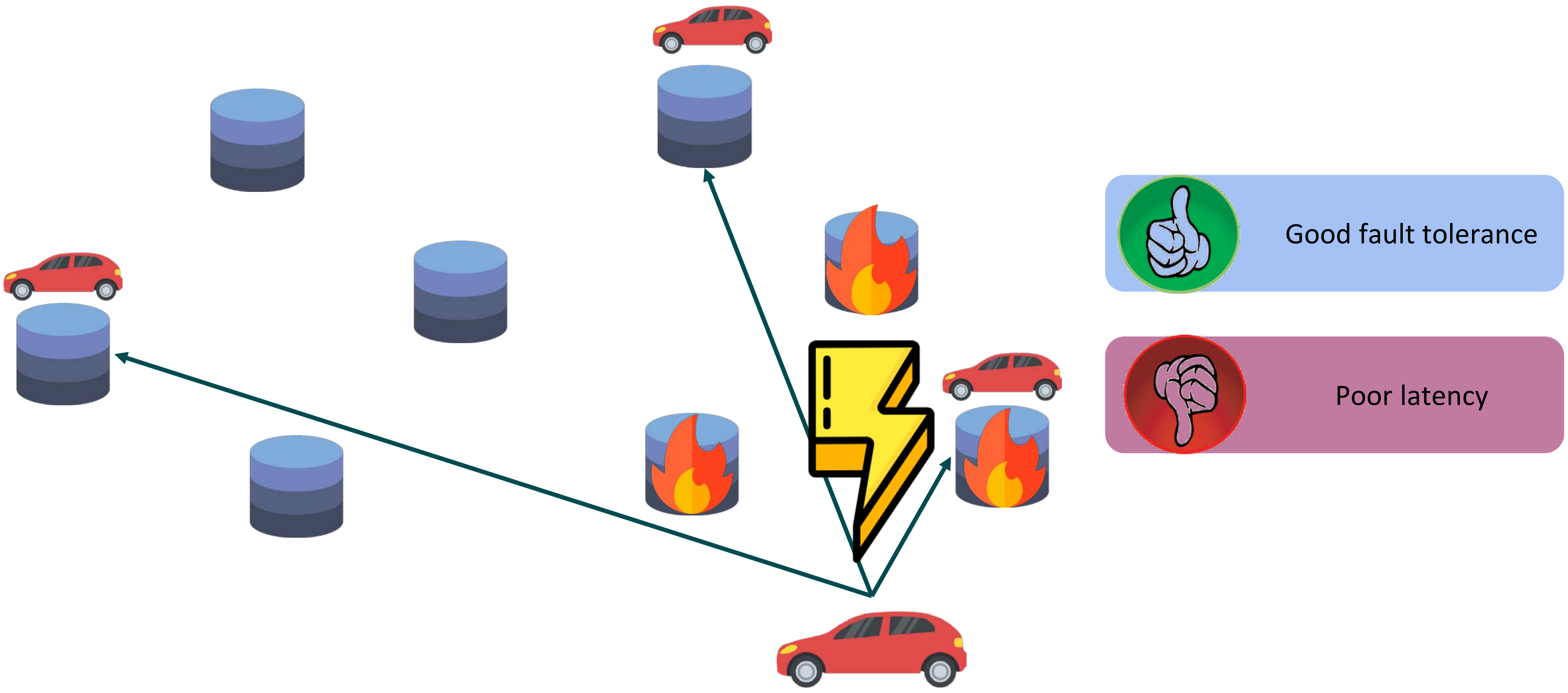


Low-latency quorum read/update

Poor fault-tolerance

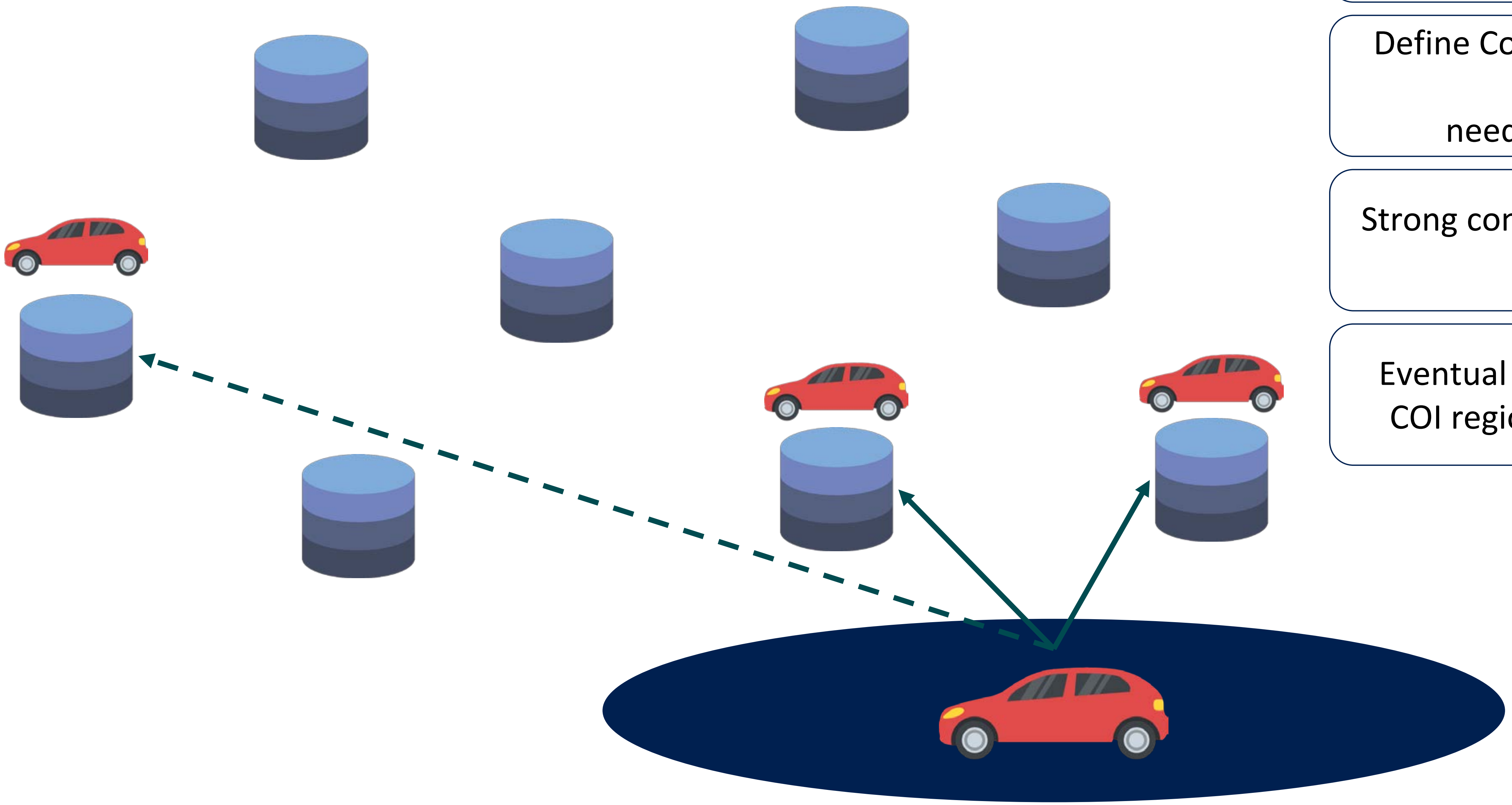
Replication: Consistency/Latency tradeoff

FogStore



Replication: Consistency/Latency tradeoff

FogStore



Utilize spatio-temporal locality nature of queries

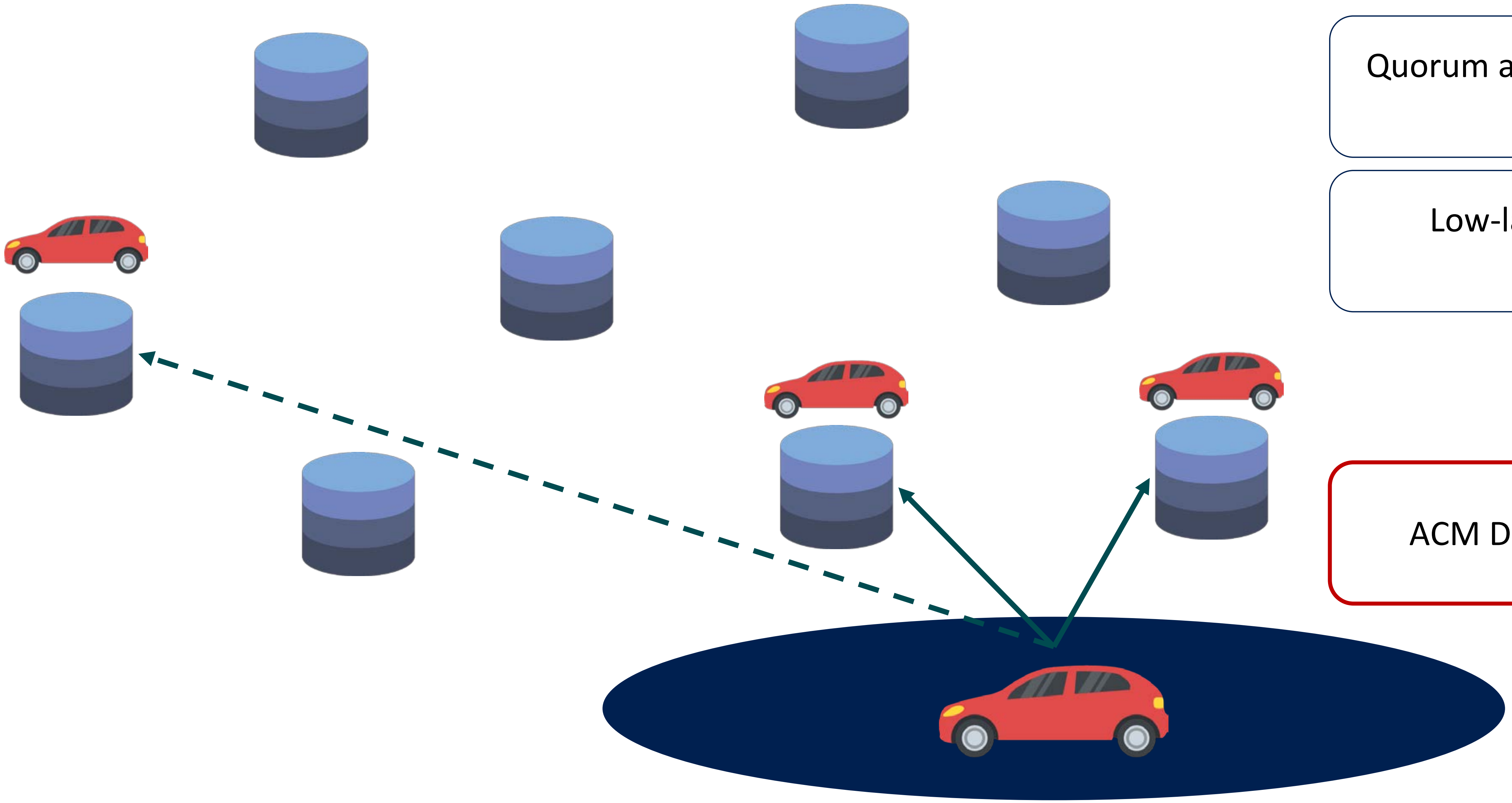
Define Context of Interest (COI) region which needs consistent data

Strong consistency only to clients in COI

Eventual consistency for out of COI region for fault-tolerance

Replication: Consistency/Latency tradeoff

FogStore



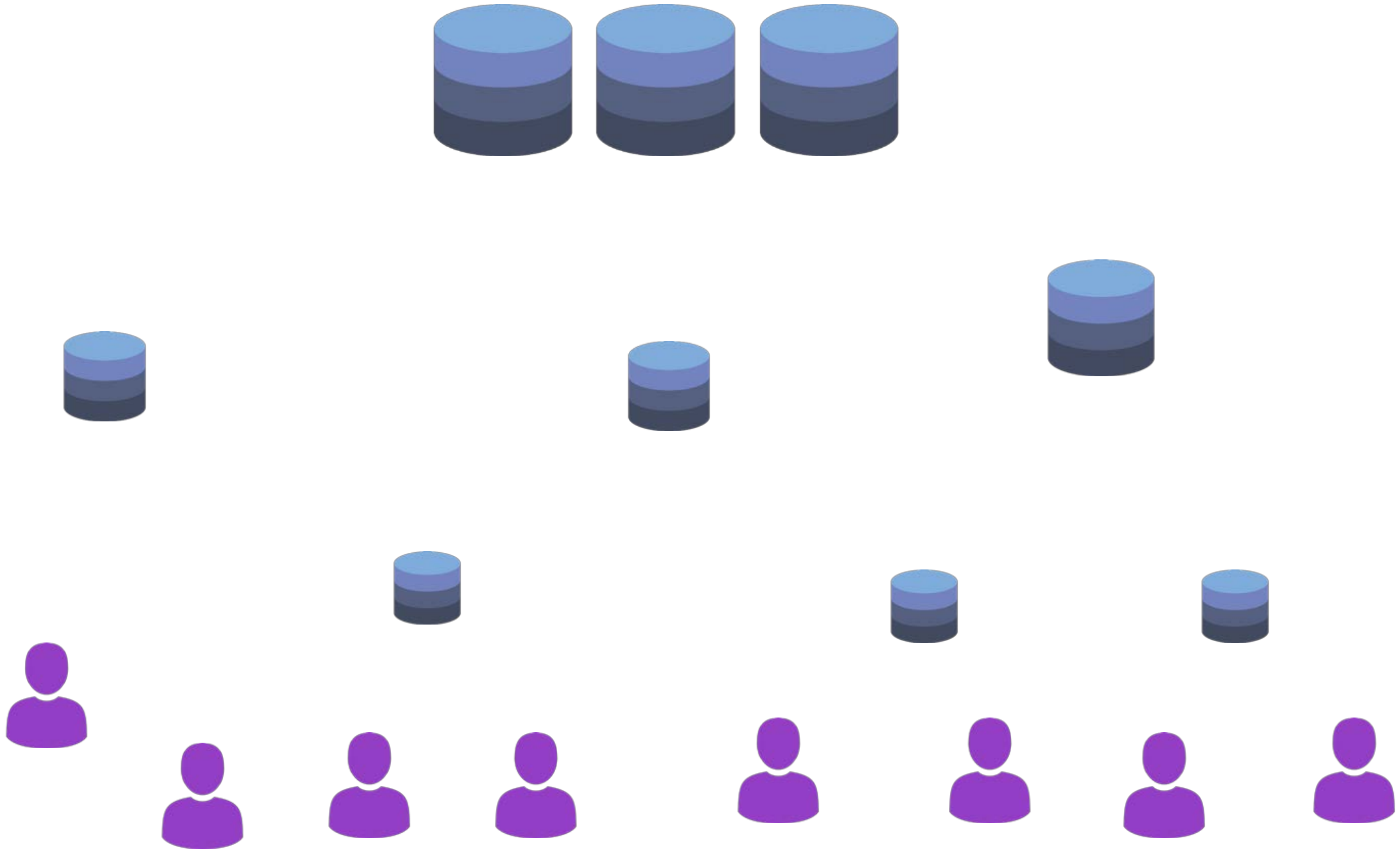
Quorum always involves replicas in proximity

Low-latency with strong consistency

ACM DEBS 2018 for details

Capacity conscious data replication

DataFog

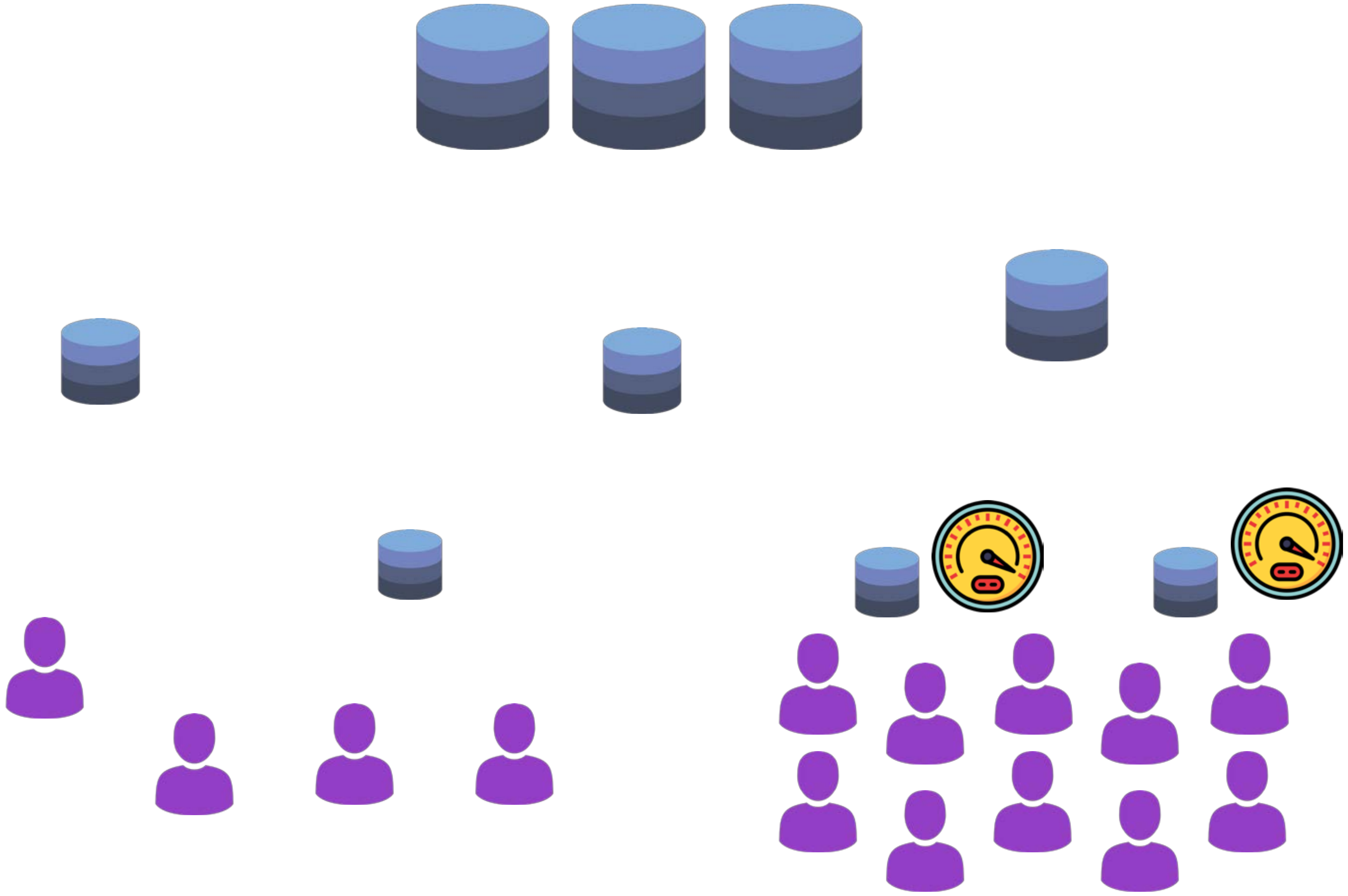


Continuous data generation

Pressure on low storage capacity of edge nodes

Capacity conscious data replication

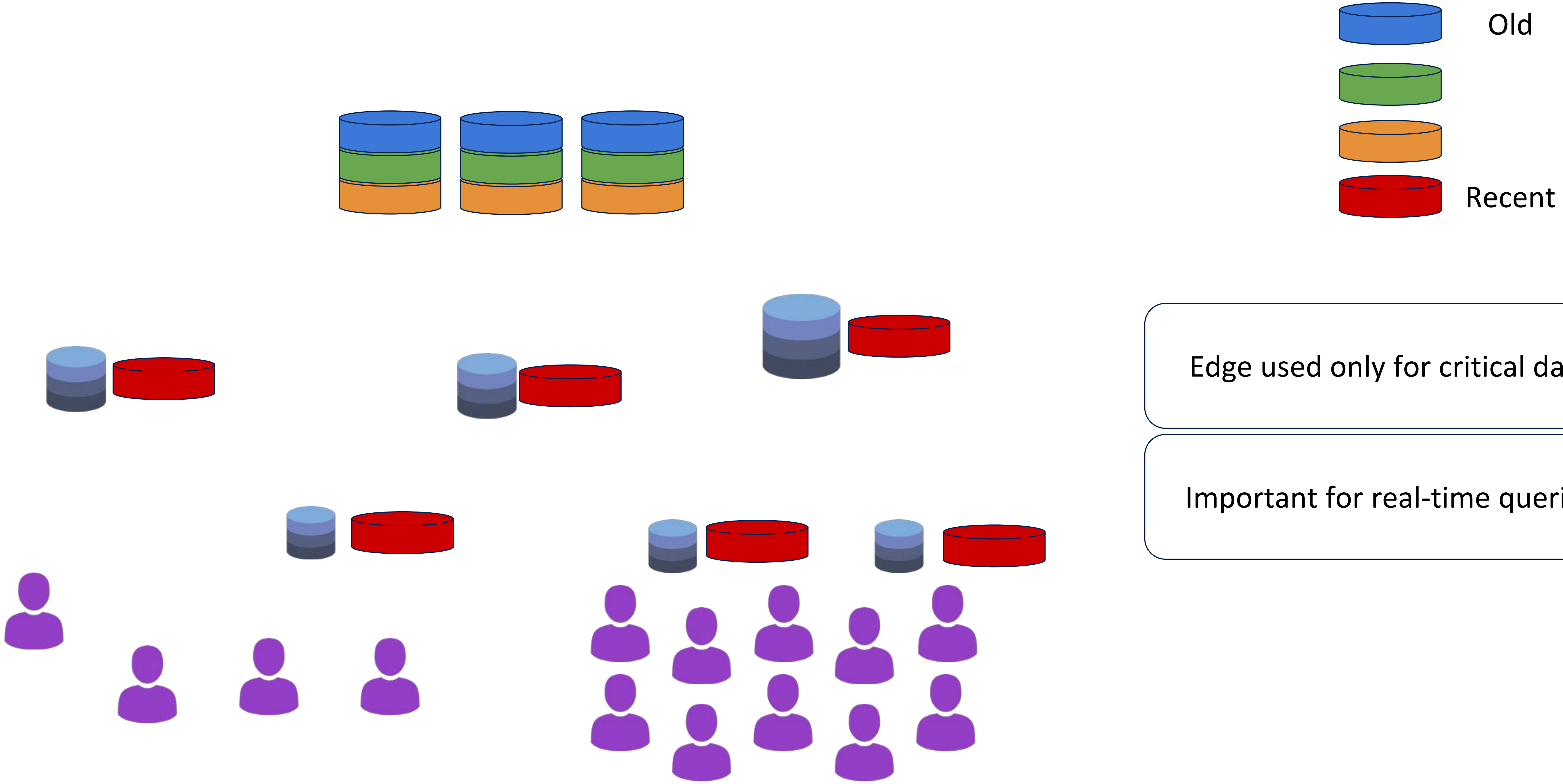
DataFog



Skews in workload distribution

Capacity conscious data replication

DataFog



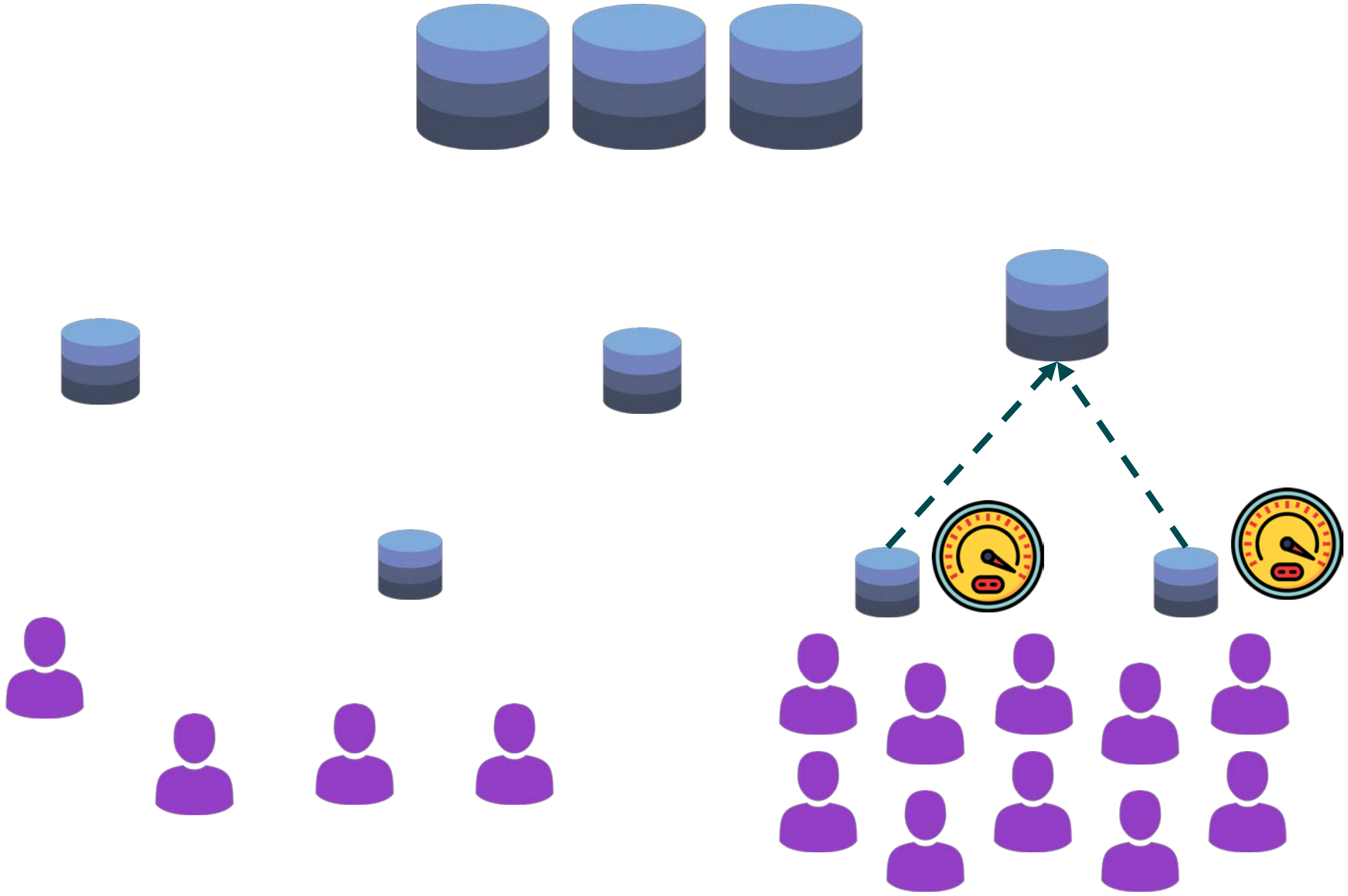
Edge used only for critical data

Important for real-time queries

Capacity conscious data replication

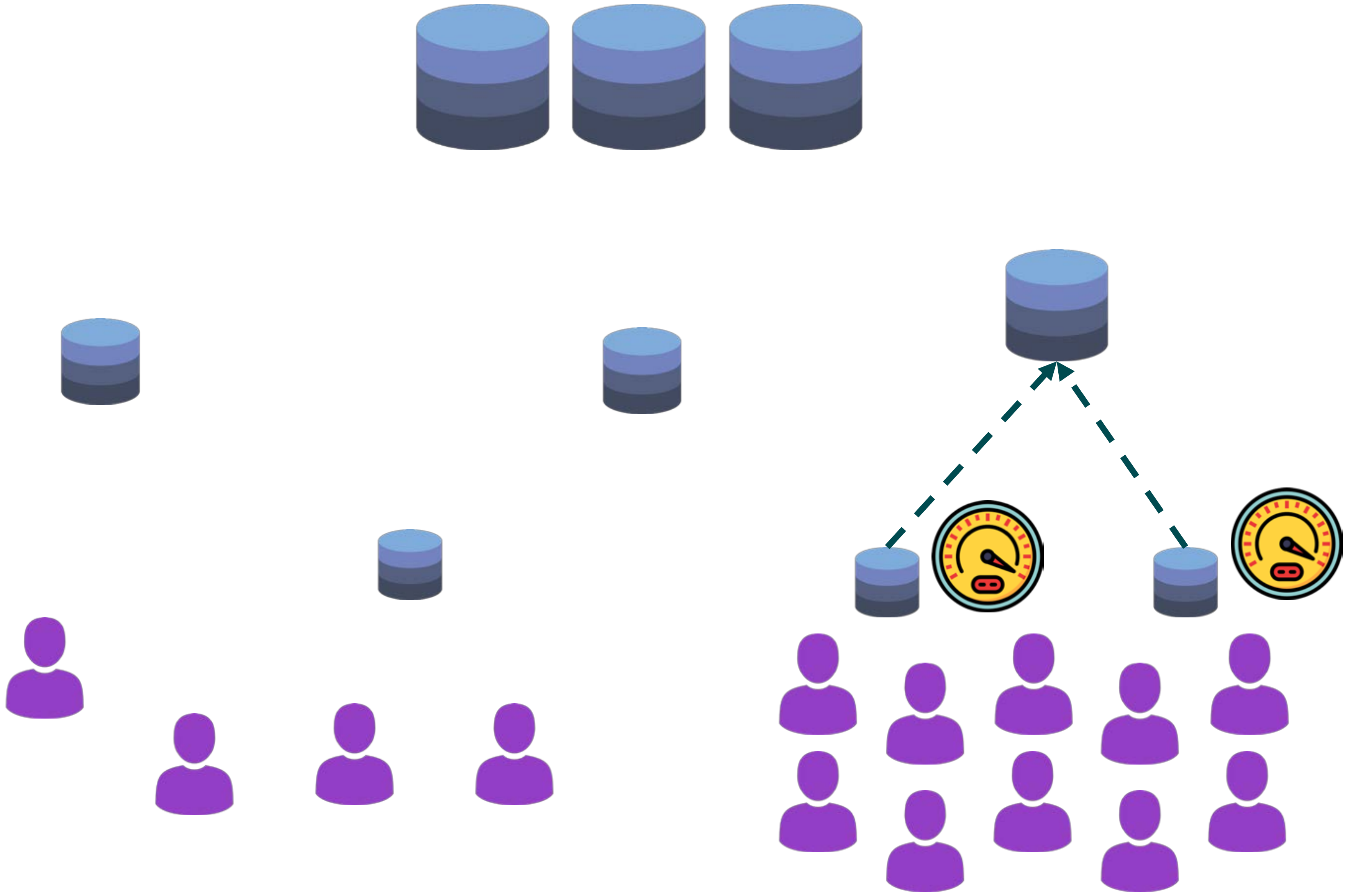
DataFog

Agile load balancing for
skew tolerance



Capacity conscious data replication

DataFog



Agile load balancing for skew tolerance

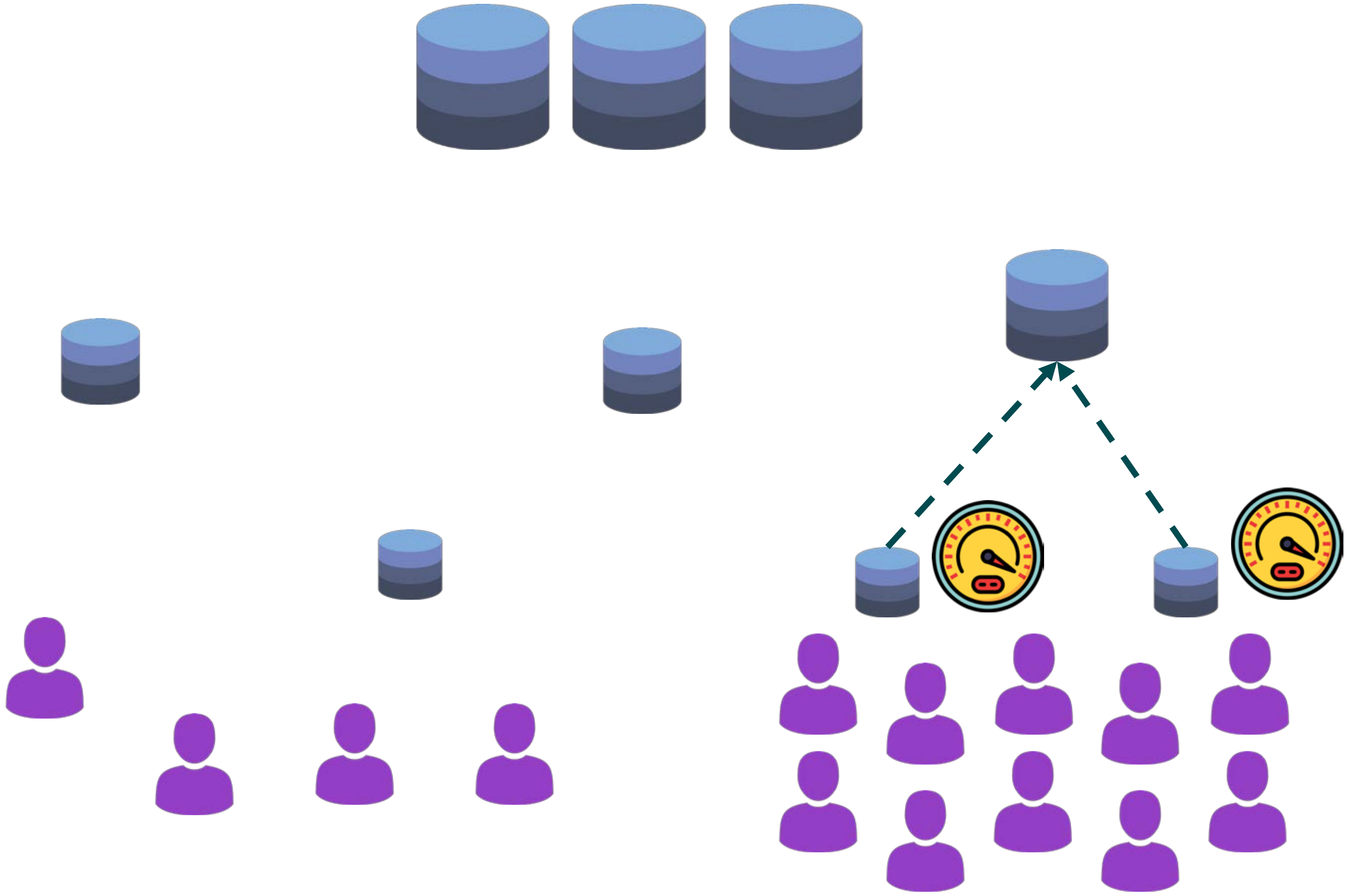
- Data-items are indexed based on their spatio-temporal attributes (e.g., Geohash)
- Consistent hashing for the location, timestamp and item-type attributes is used for partitioning data across nodes
- Multiple replicas on Edge nodes for low latency
- Multiple replicas on remote datacenter nodes for tolerance from geographically correlated failures
- Mechanisms for adapting to hotspots

Capacity conscious data replication

DataFog

HotEdge 2018 for details

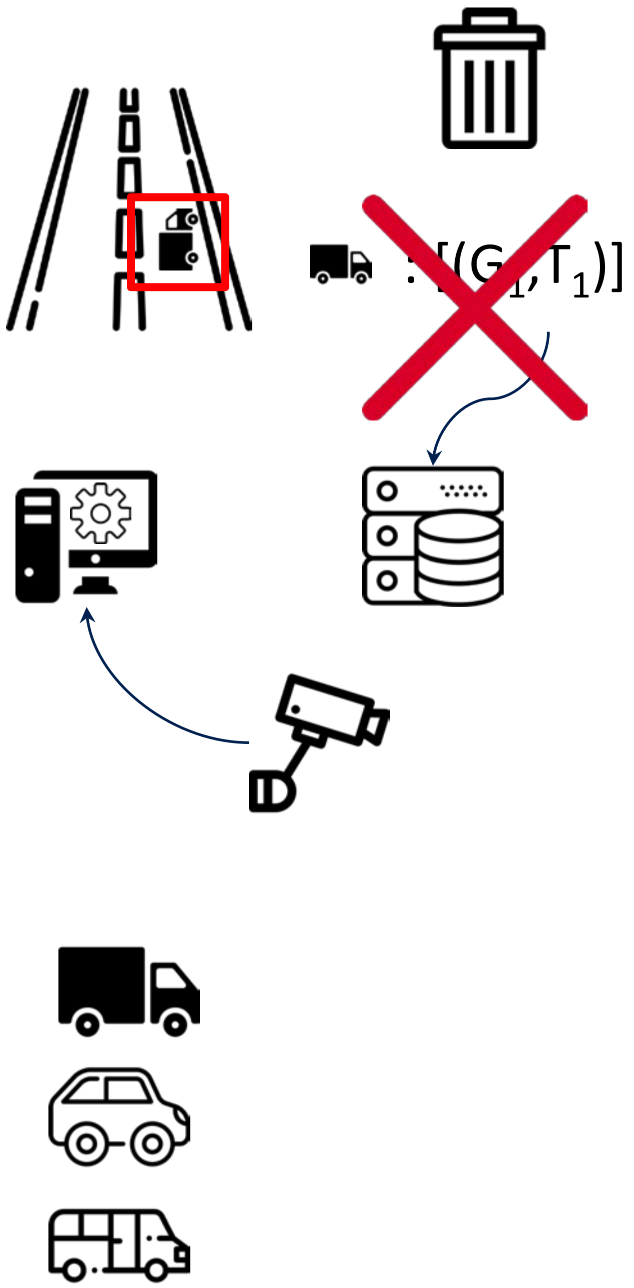
Agile load balancing for skew tolerance



- Data-items are indexed based on their spatio-temporal attributes (e.g., Geohash)
- Consistent hashing for the location, timestamp and item-type attributes is used for partitioning data across nodes
- Multiple replicas on Edge nodes for low latency
- Multiple replicas on remote datacenter nodes for tolerance from geographically correlated failures
- Mechanisms for adapting to hotspots

Applications using Autonomous Edge

Space Time Trajectory Registration (ST



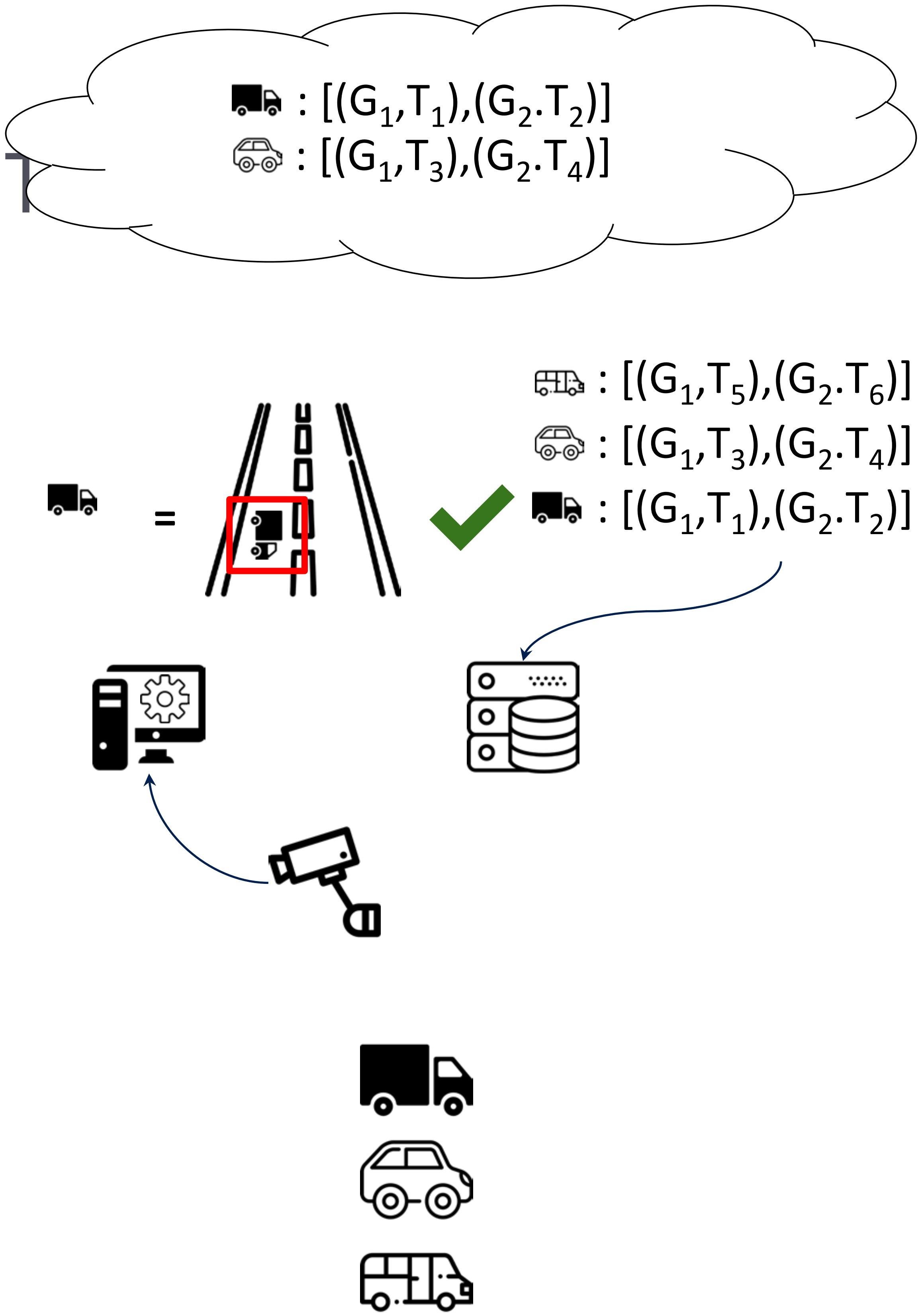
ACM DEBS 2018 for details

Forward/Backward propagation between cameras

Greedy/Lazy trajectory aggregation

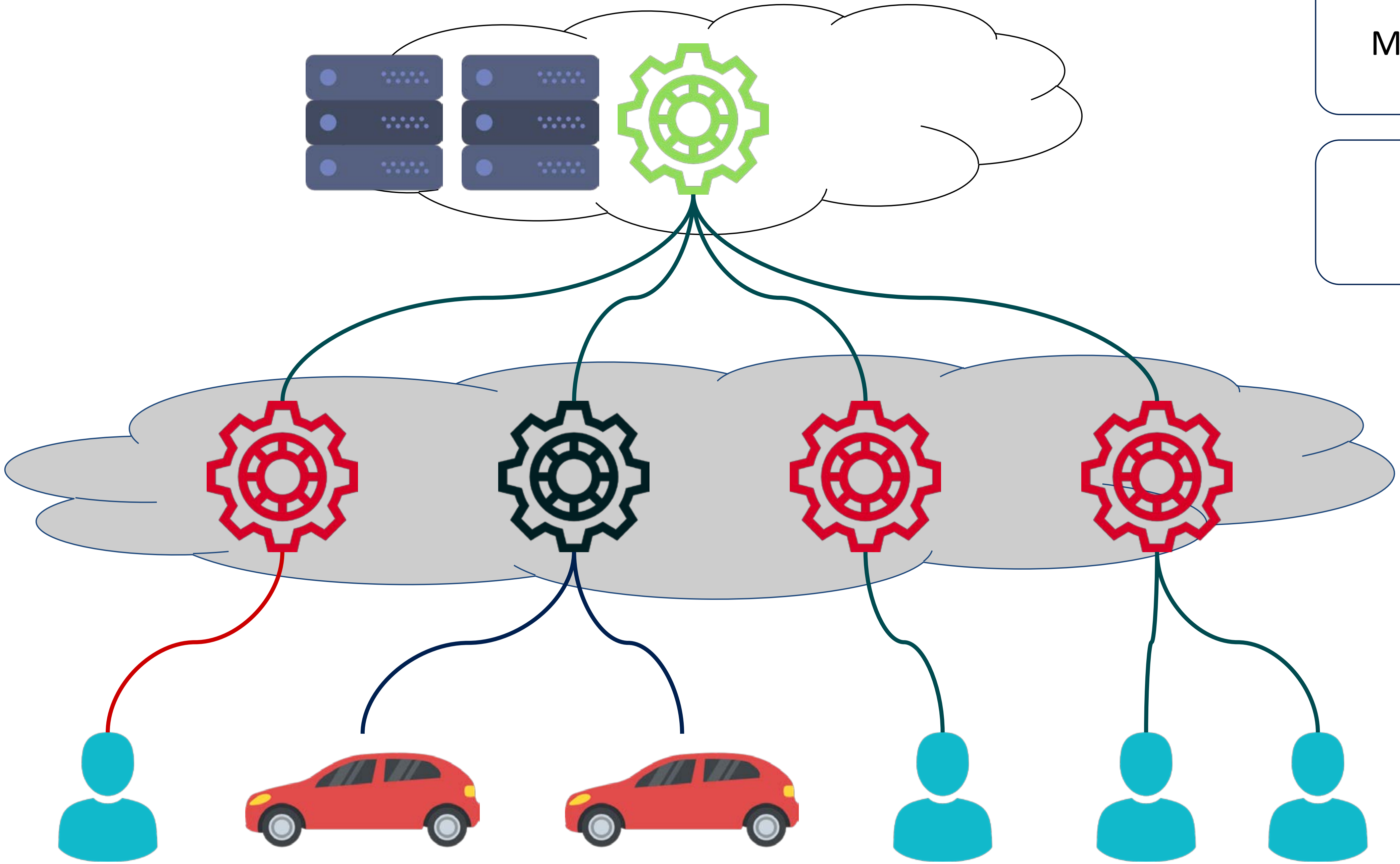
Storage is bounded by the activities within each camera

Edge as real-time processing
Cloud as history



Applications using Autonomous Edge

Social sensing

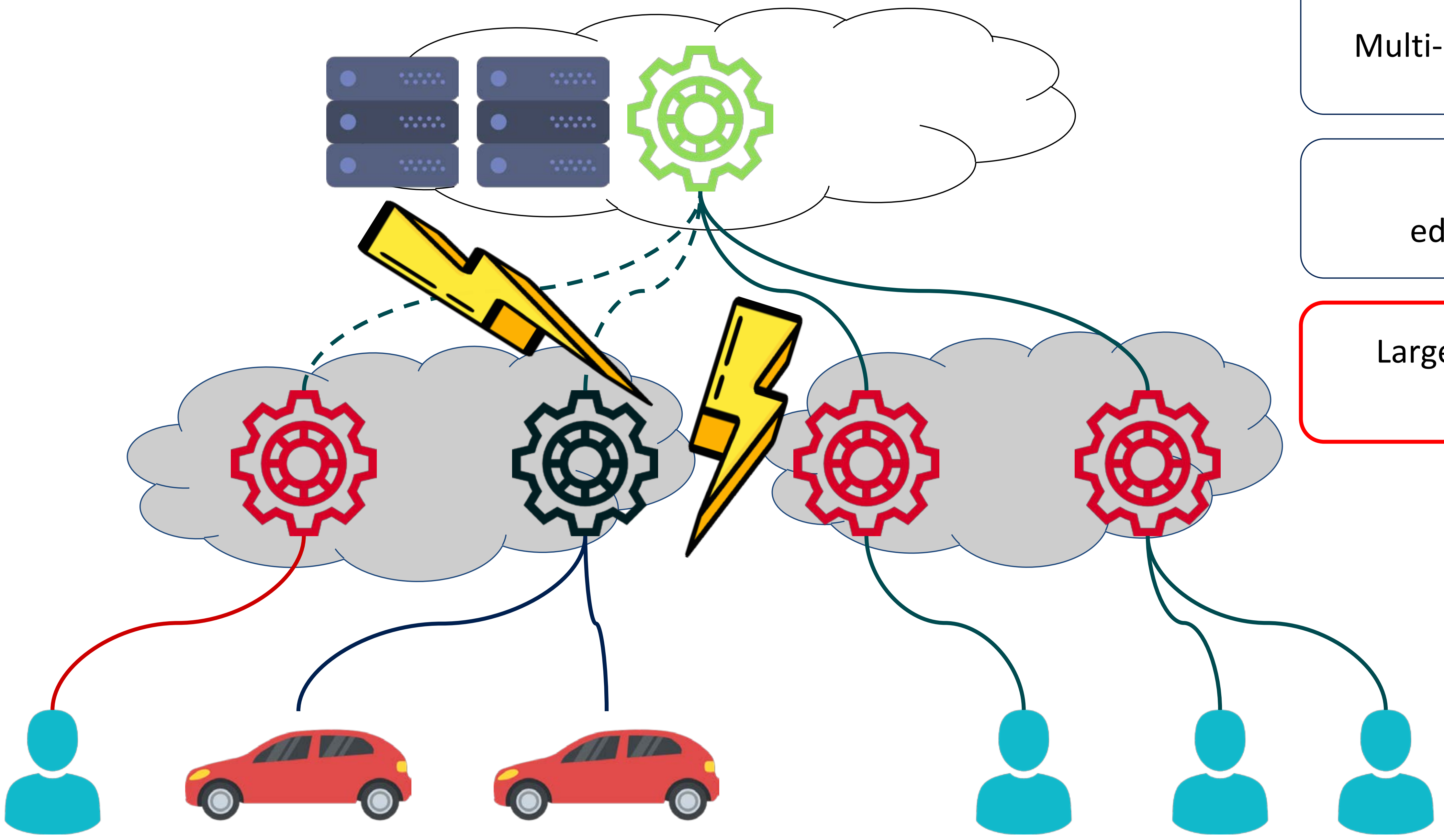


Multi-component applications

Deployed across
edge-cloud continuum

Applications using Autonomous Edge

Social sensing



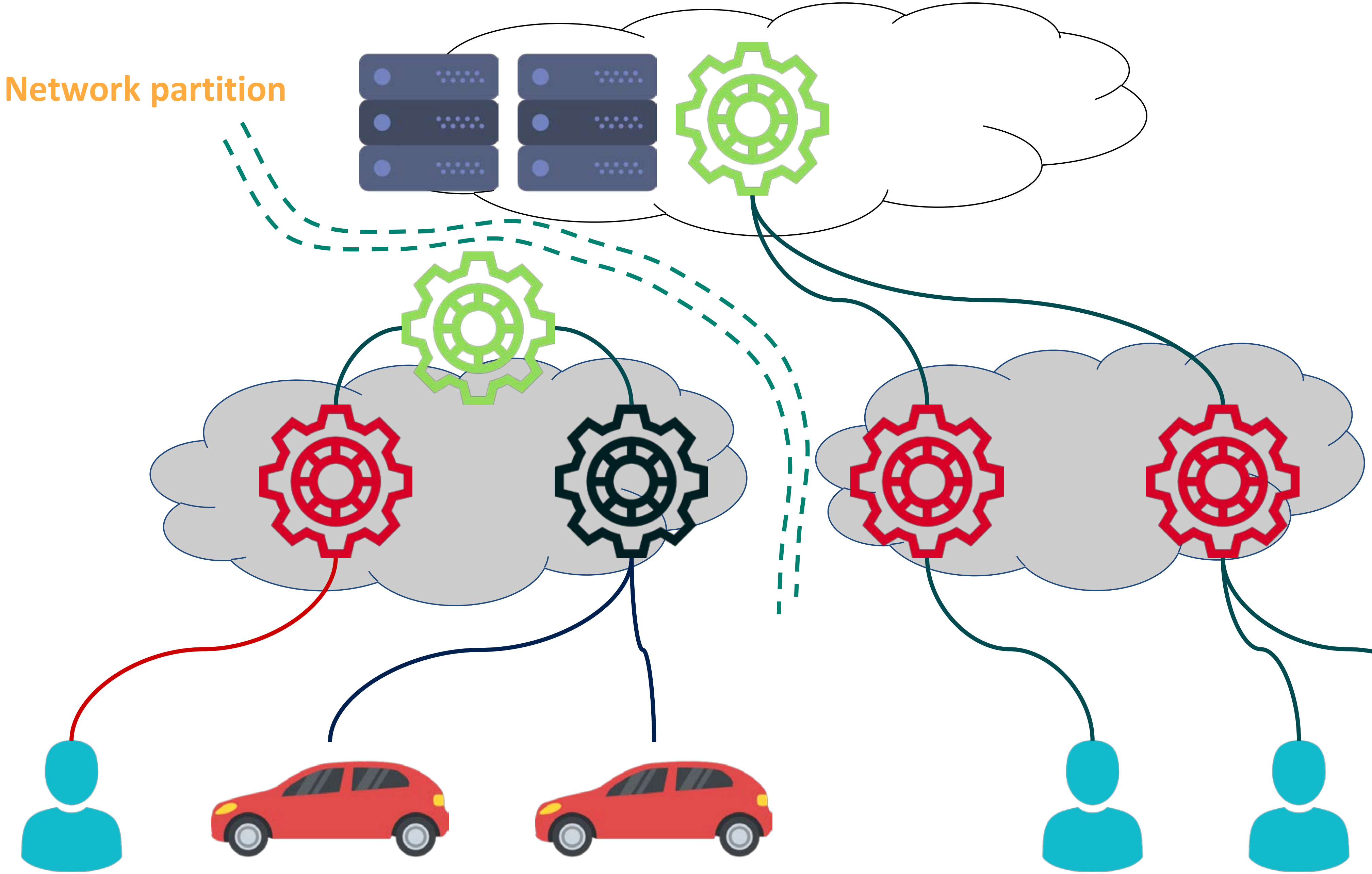
Multi-component applications

Deployed across
edge-cloud continuum

Large-scale outages lead to
connectivity loss

Applications using Autonomous Edge

Social sensing



Multi-component applications

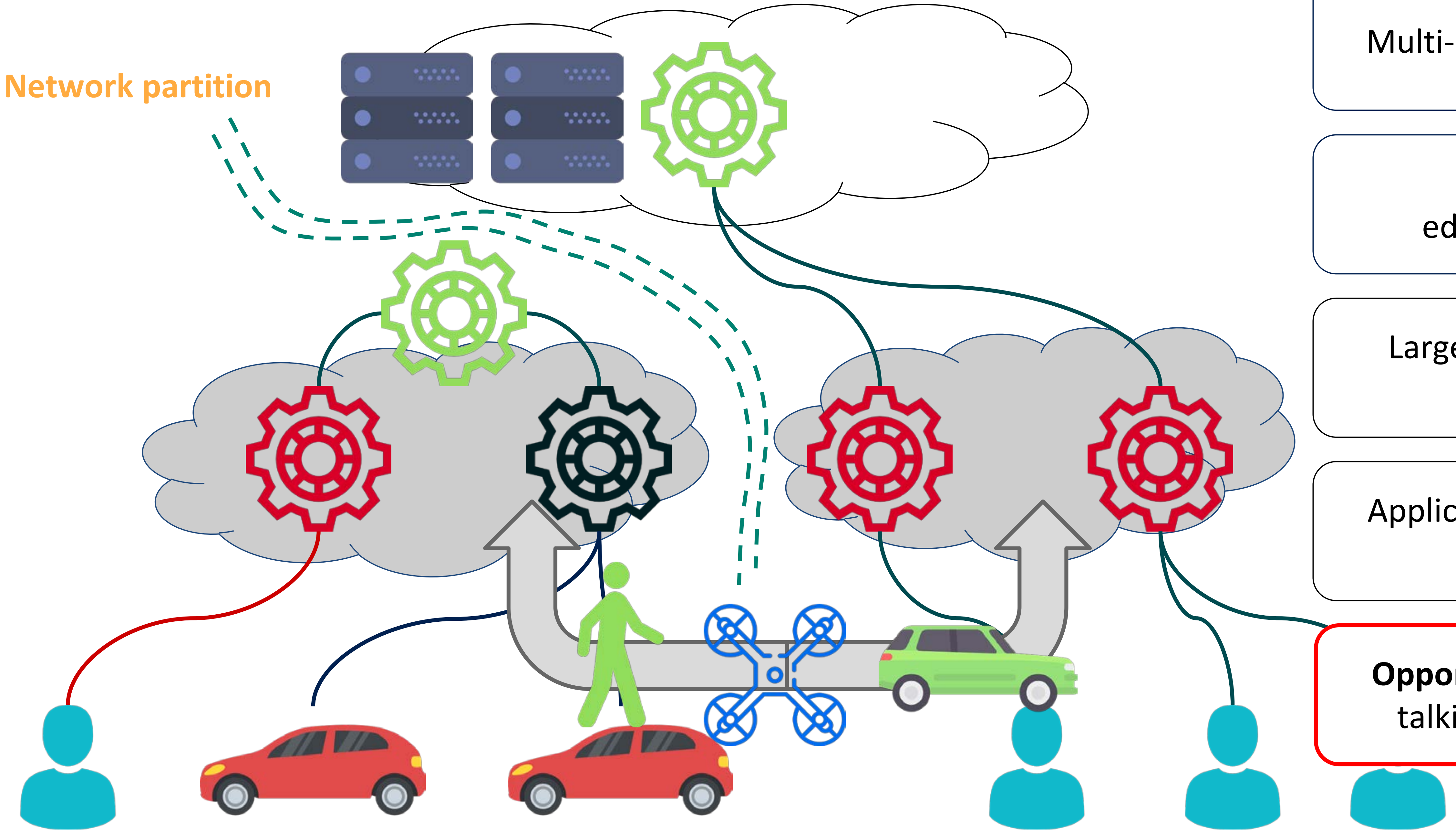
Deployed across
edge-cloud continuum

Large-scale outages lead to
connectivity loss

Application reconfiguration to
resume operation

Applications using Autonomous Edge

Social sensing



Multi-component applications

Deployed across edge-cloud continuum

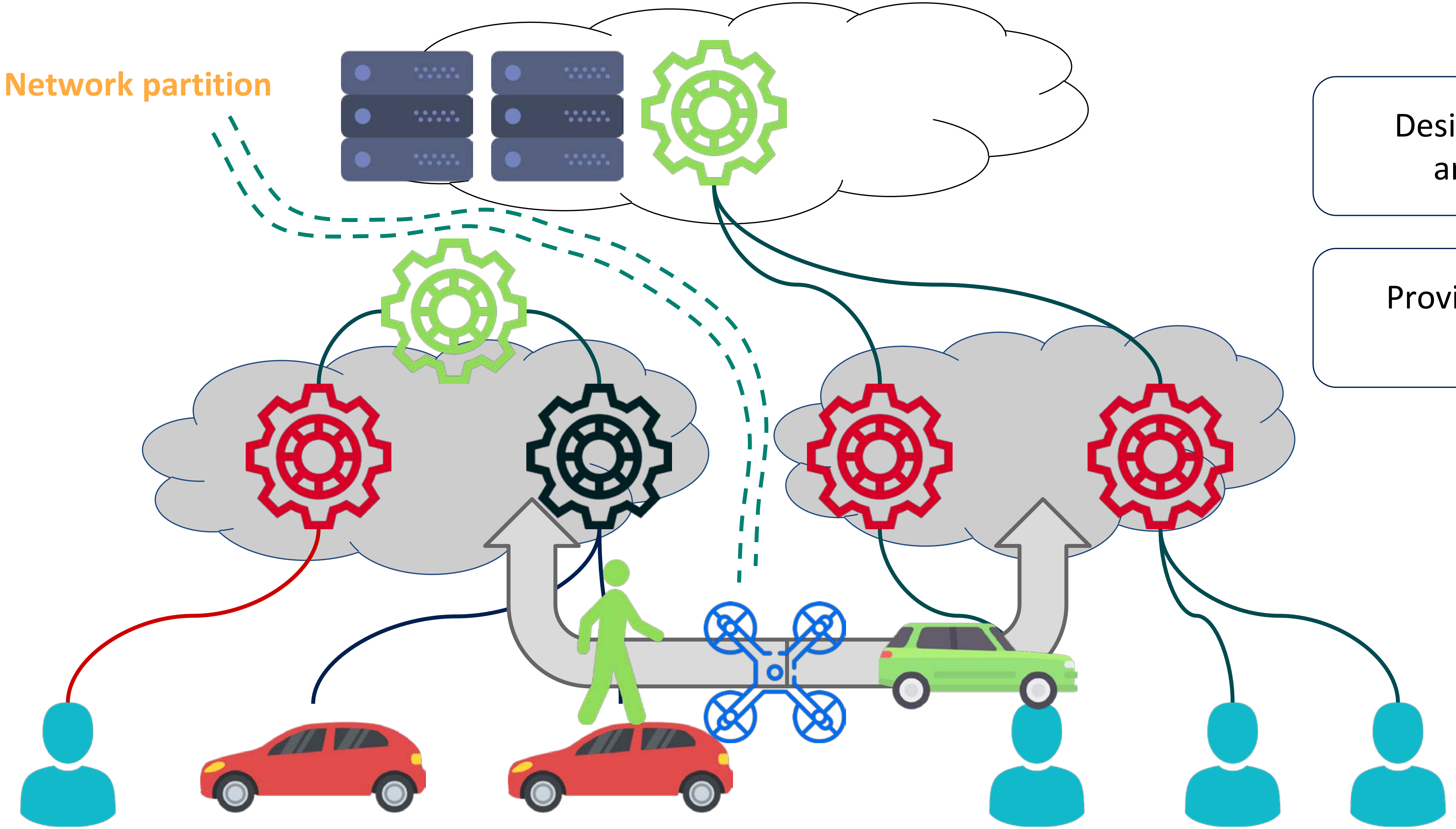
Large-scale outages lead to **connectivity loss**

Application reconfiguration to resume operation

Opportunistic networking for talking between partitions

Applications using Autonomous Edge

Social sensing



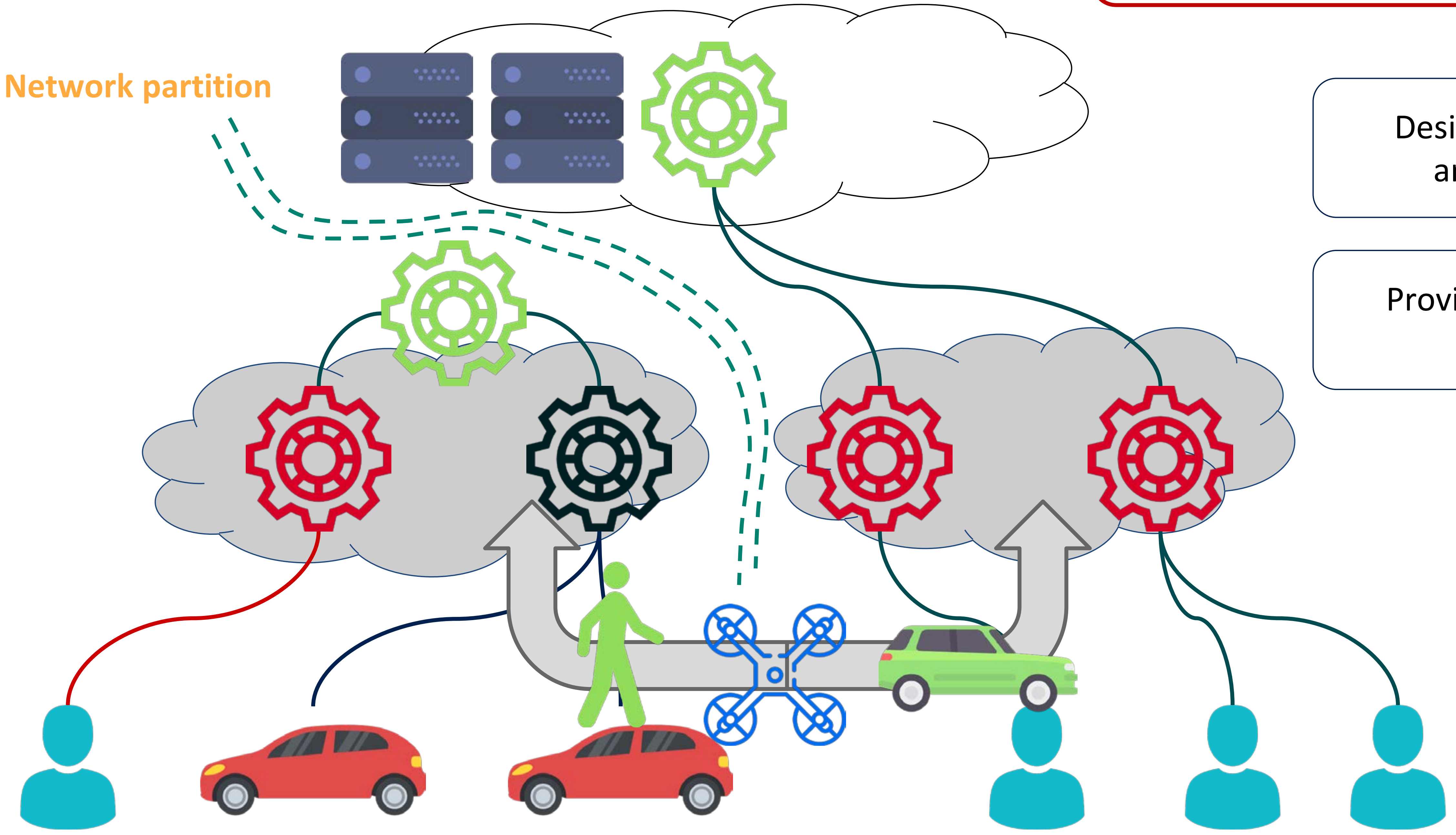
Design a generic multi-tier architecture for apps

Provide APIs for developing application logic

Applications using Autonomous Edge

Social sensing

SocialSens 2017 for details

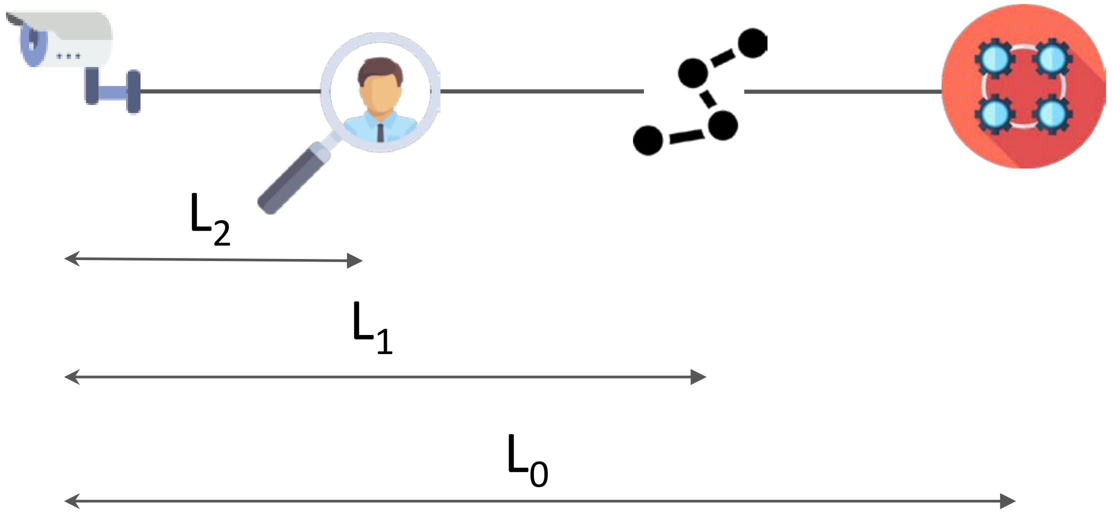


Design a generic multi-tier architecture for apps

Provide APIs for developing application logic

Ongoing work: Logically centralized control plane

1. Extension of Foglets programming model to add QoS requirements
   - Max data staleness at each level
2. Centralized control for end-to-end allocation respecting SLAs
3. Enables high level resource management policies
   - E.g. resource consolidation for energy minimization



Ongoing work: Logically centralized control plane

1. How distributed can the control plane be?
    - Tradeoff between control plane latency and end-to-end decision making
2. How to efficiently monitor vastly geo-distributed resources?
    - Necessary for adaptive reconfigurations
    - Devise decentralized monitoring schemes
    - Piggyback on data plane
3. How to deal with inconsistent resource state at control plane?
    - Controller's world view may be stale due to failures

Concluding Remarks

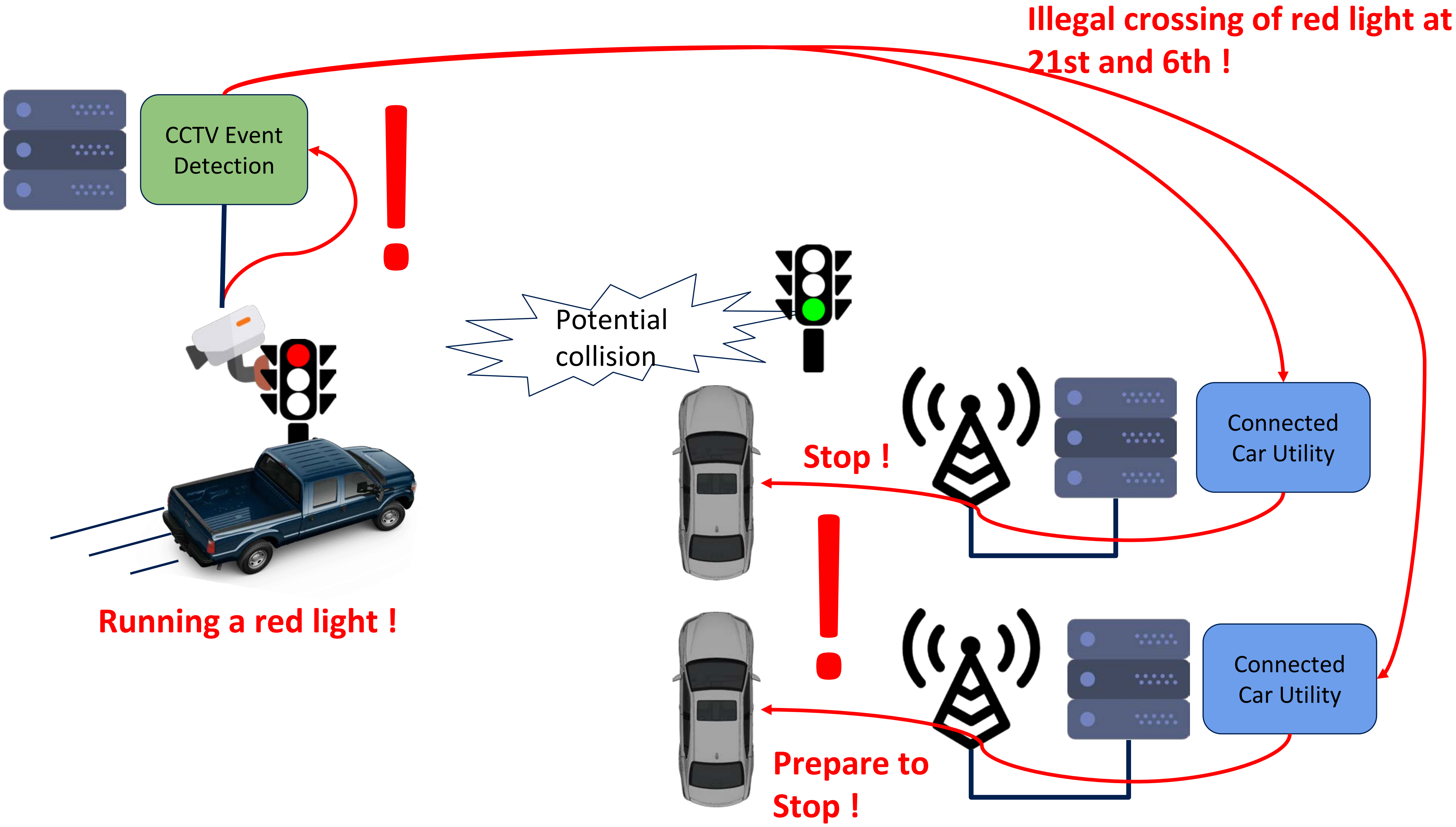
- Inflection point in systems research spurred by large-scale deployment of sensors and novel situation awareness applications
- Edge/Fog emerging as a serious disruption to the Cloud status quo
- Vision for the future

Questions?

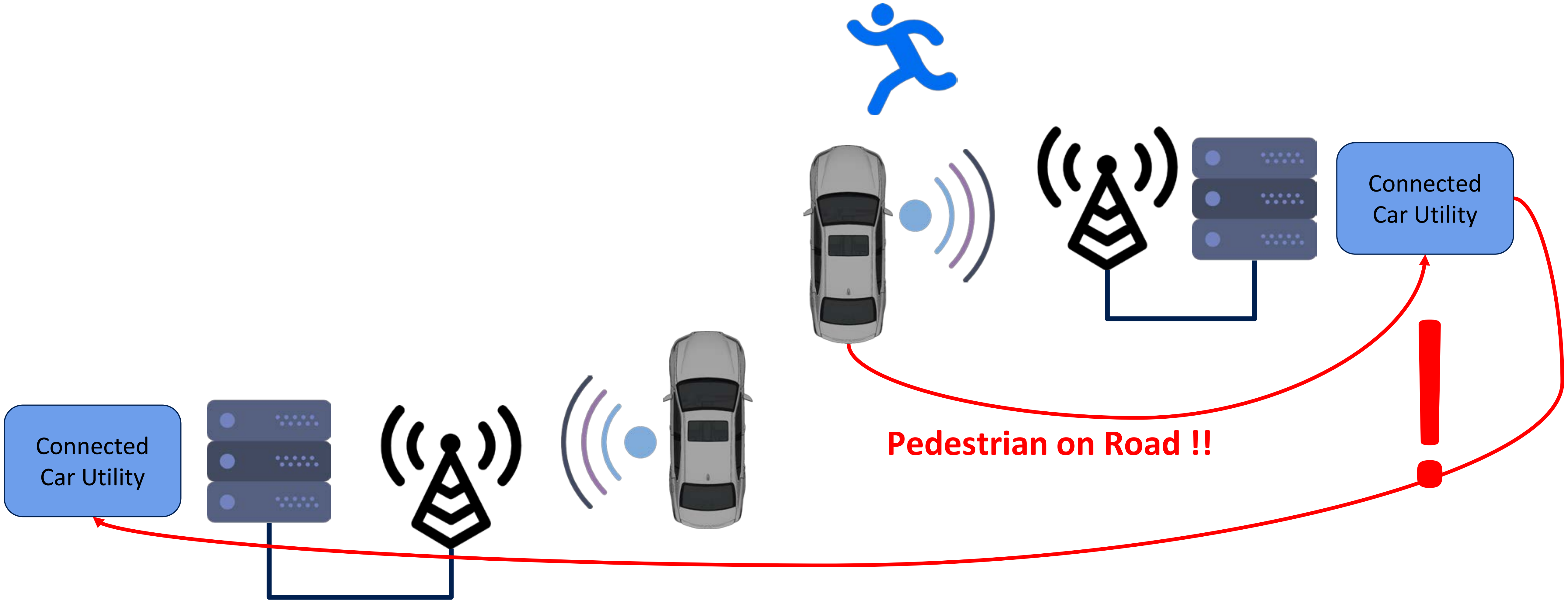
Horizontal communication

Why horizontal communication across edge nodes ?

- Can't expect multiple interacting entities to be connected to the same edge node at a time
  - Assumption : Each cell tower (eNB) has an edge-cluster
  - A vehicle connects to edge-cluster on the cell tower it's connected to
  - Cell tower (eNB) selection done locally based on best SNR
  - Two clients very close-by may be connected to different eNBs
- Allows a more flexible model , wherein low-latency messaging is provided not just to clients connected to same eNB
- In future networks, the size of base stations is going to become smaller (small cells in 5G), which would require more cross-base-station communications
- Avoiding redundancy : Nearby edge nodes share context, and making them independent would mean increased redundancy in their actions
- Load balancing : Hotspot formation is much more likely if each edge node works in isolation. P2P communication needed for better load balancing.

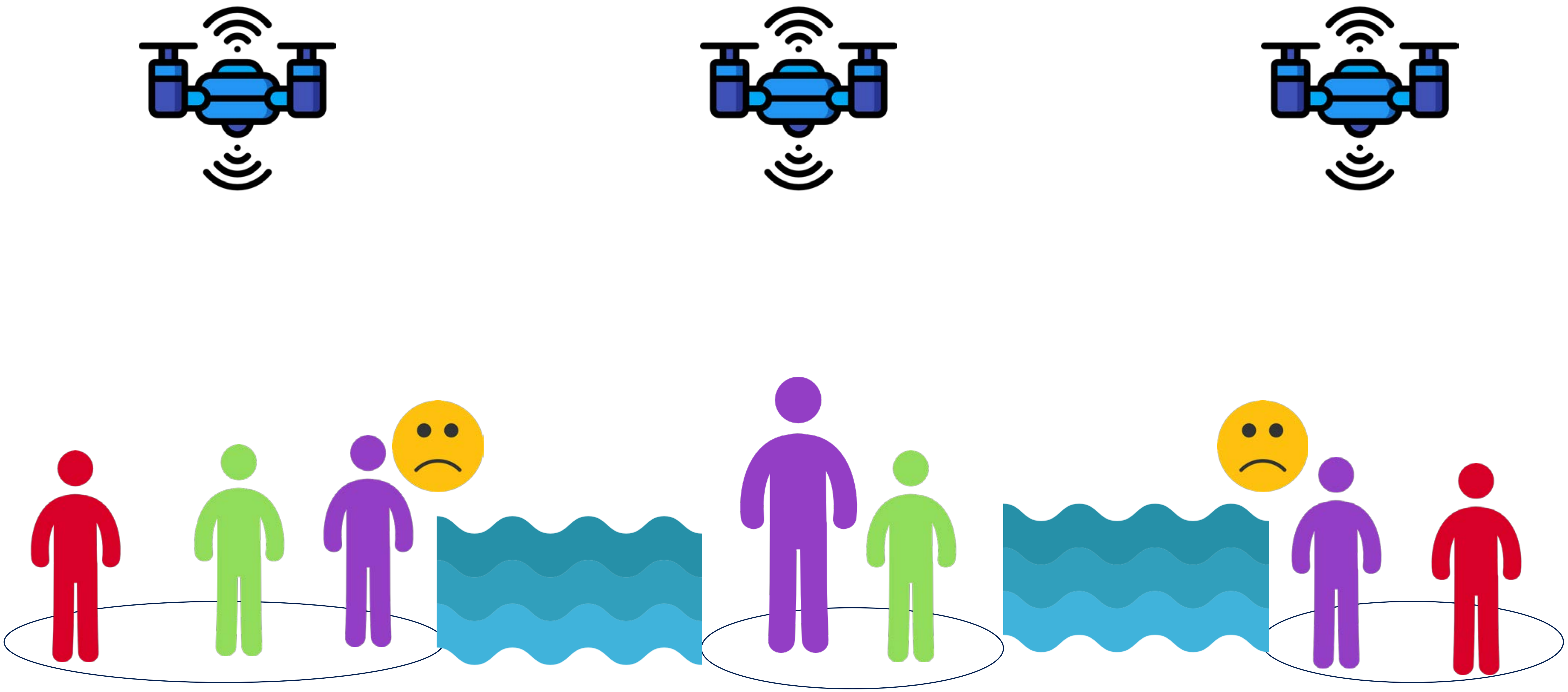

Illegal crossing of red light at 21st and 6th !
CCTV Event Detection
Potential collision
Running a red light !
Stop !
Connected Car Utility
Connected Car Utility
Prepare to Stop !

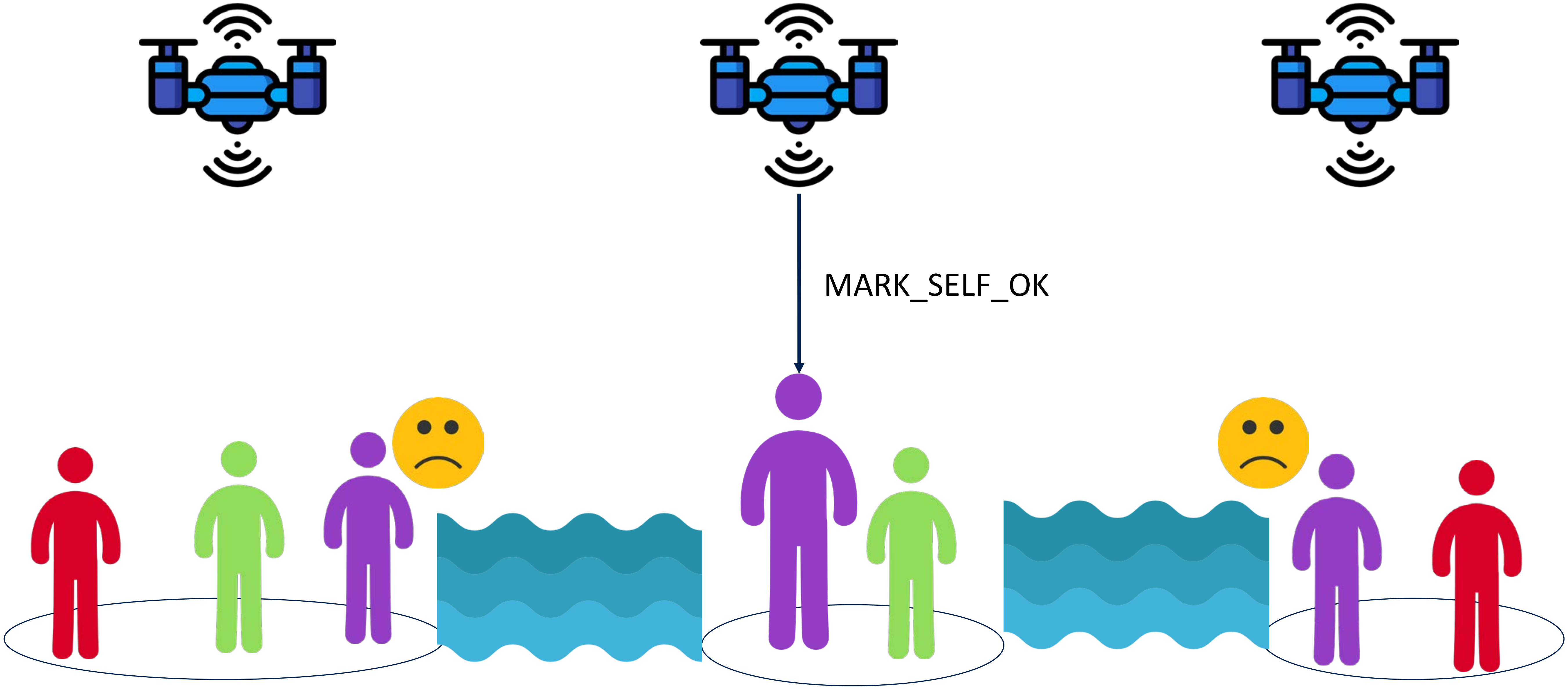
In worst case scenario, the object detection technology on each car would detect this jay-walking pedestrian. Proactive alerts are necessary to avoid such situations.

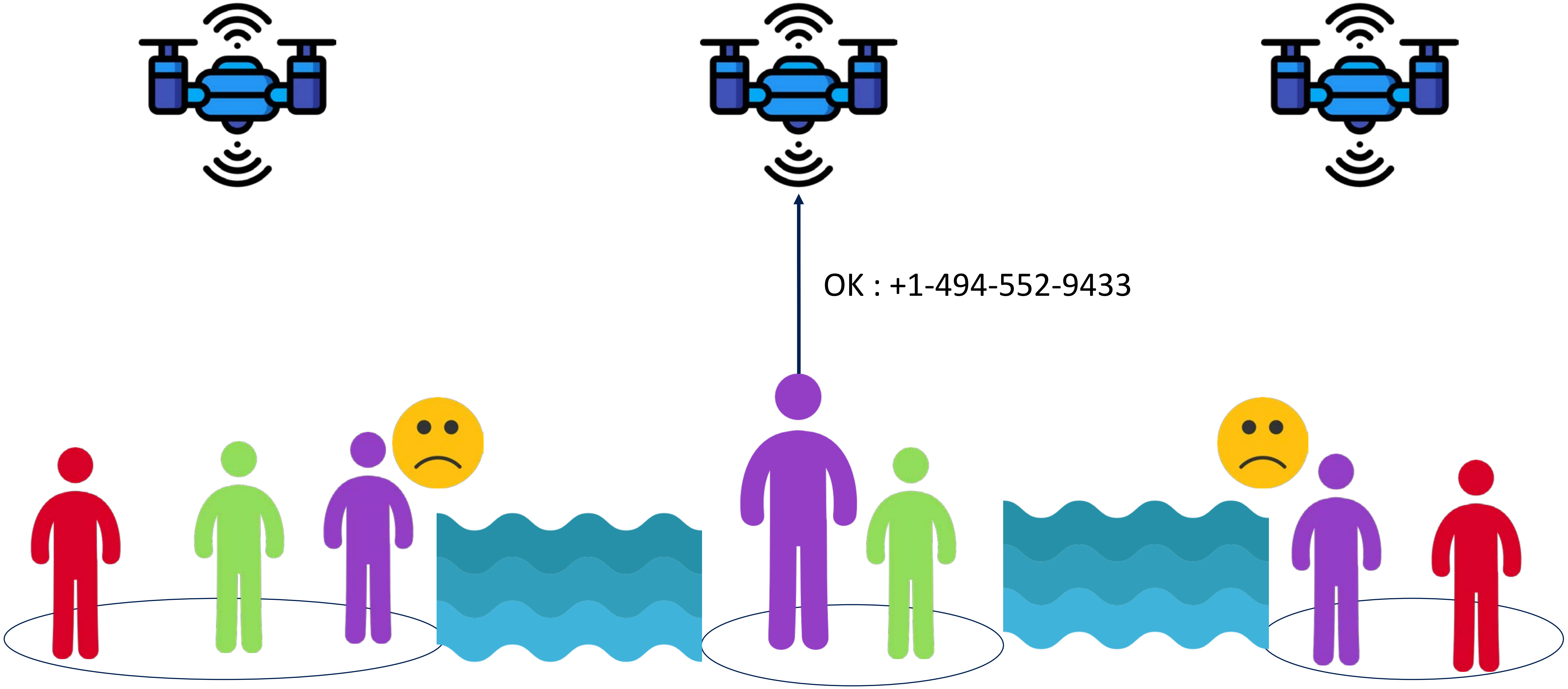


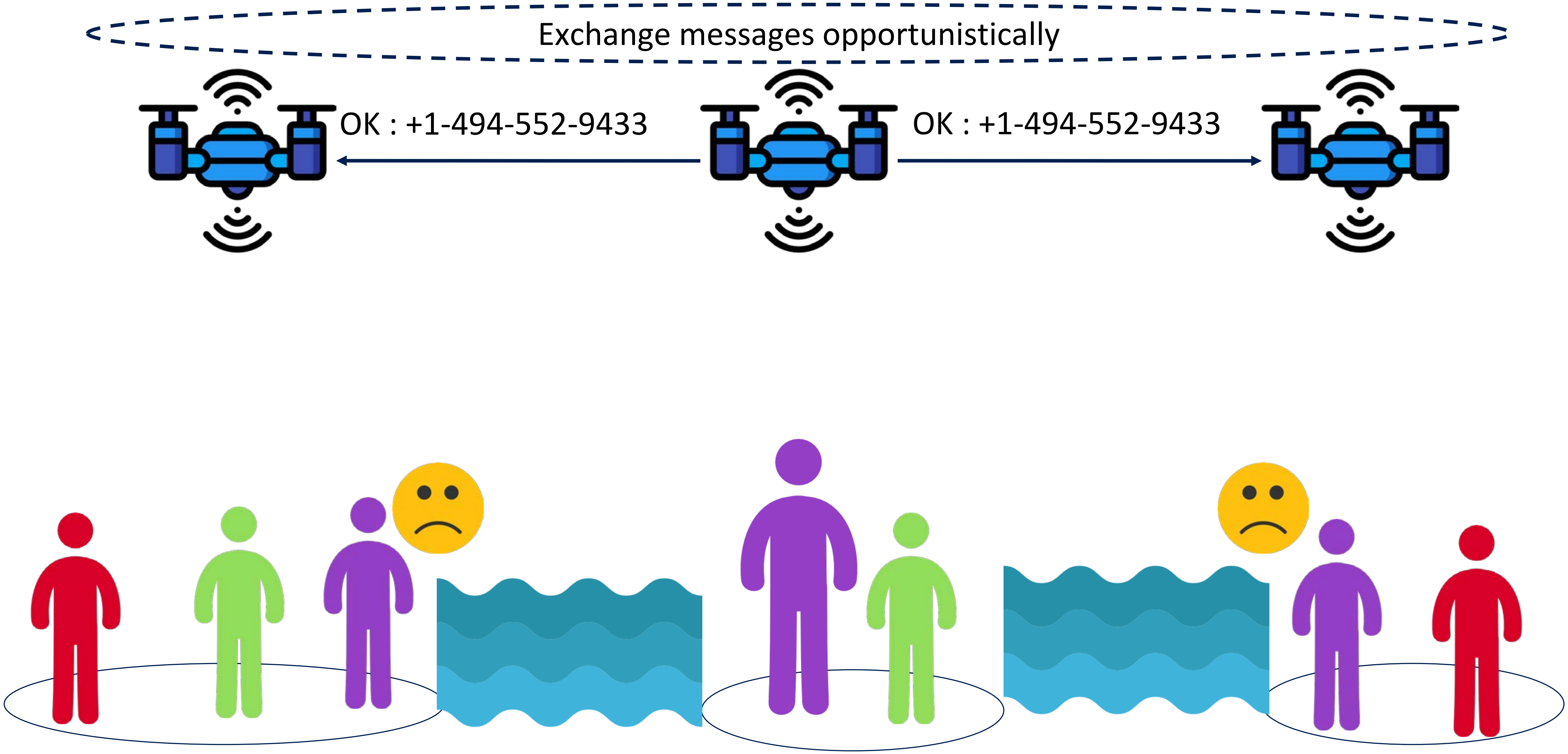
Edge without cloud

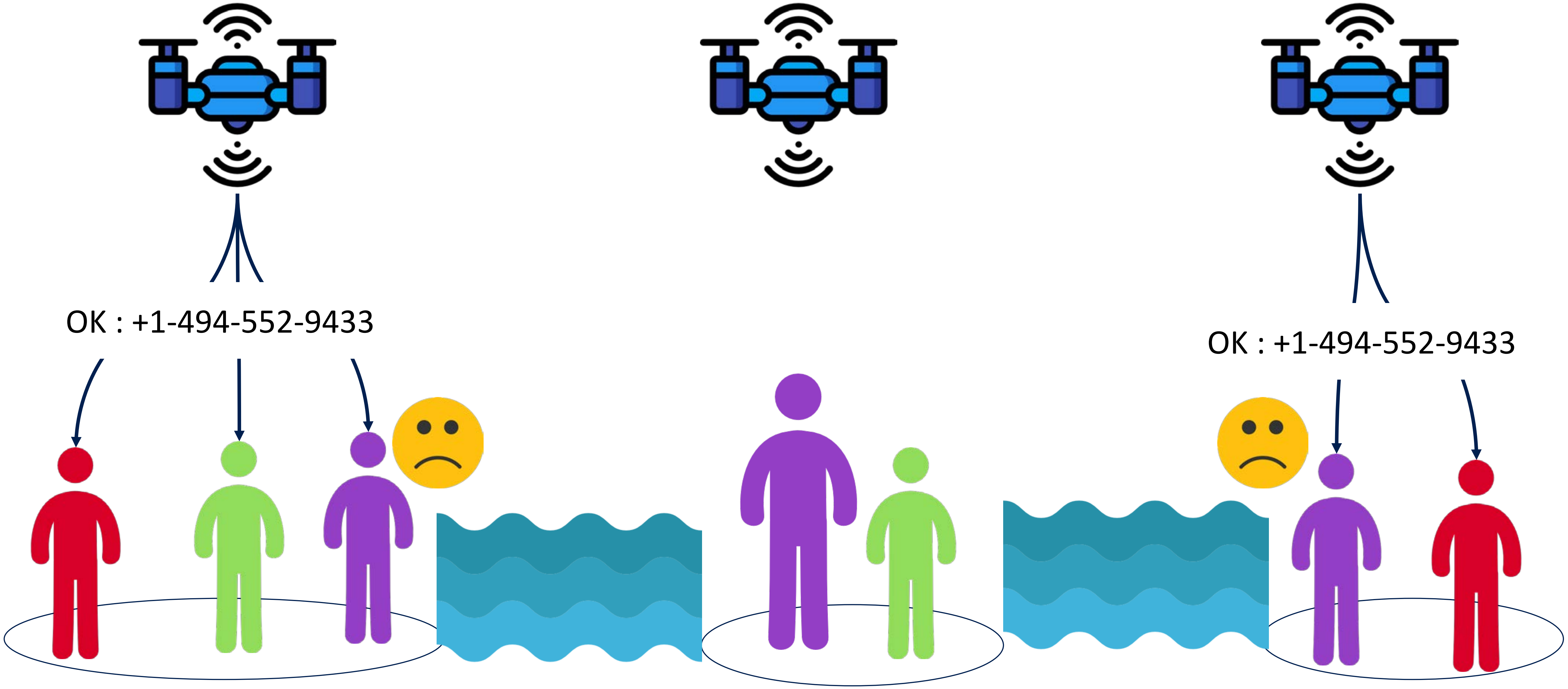
People of the same colour are 1 family. Concerned about other family members

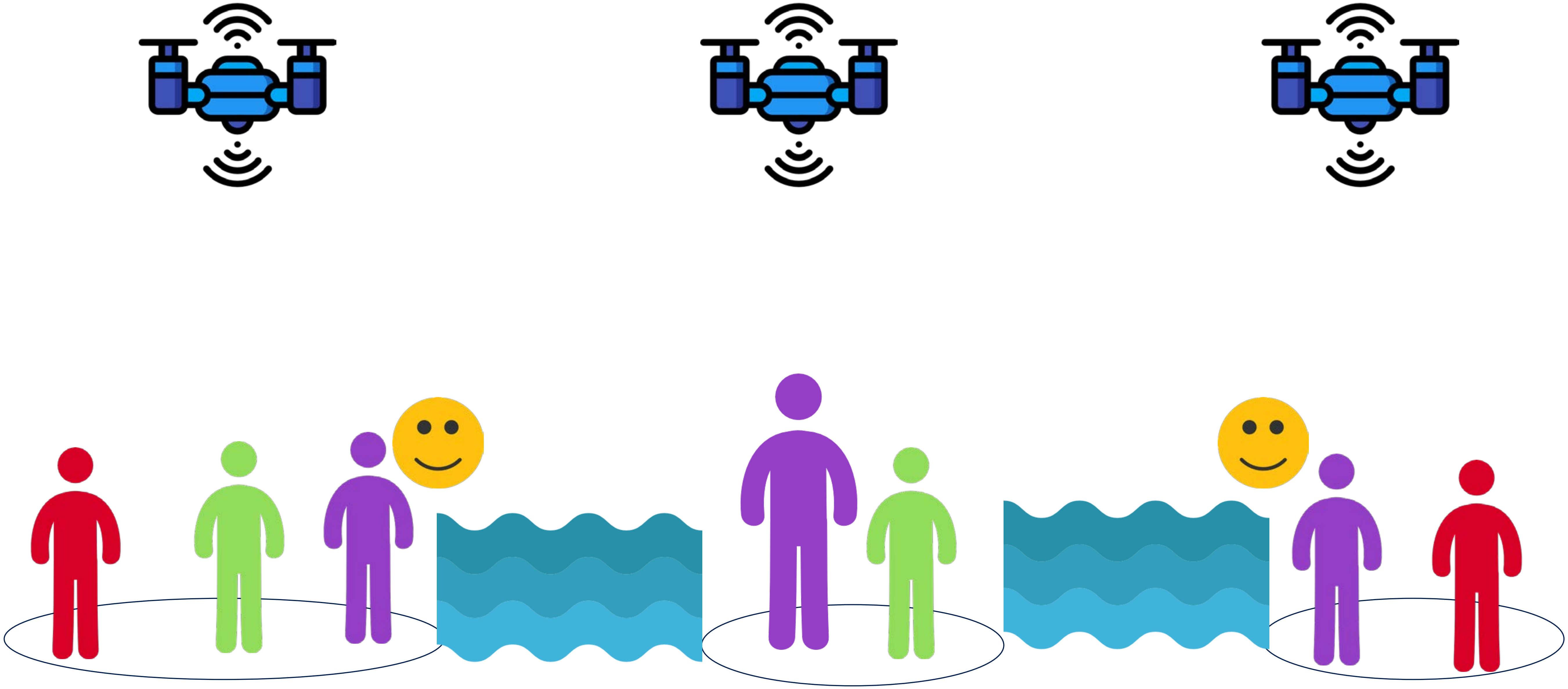



MARK_SELF_OK


OK : +1-494-552-9433

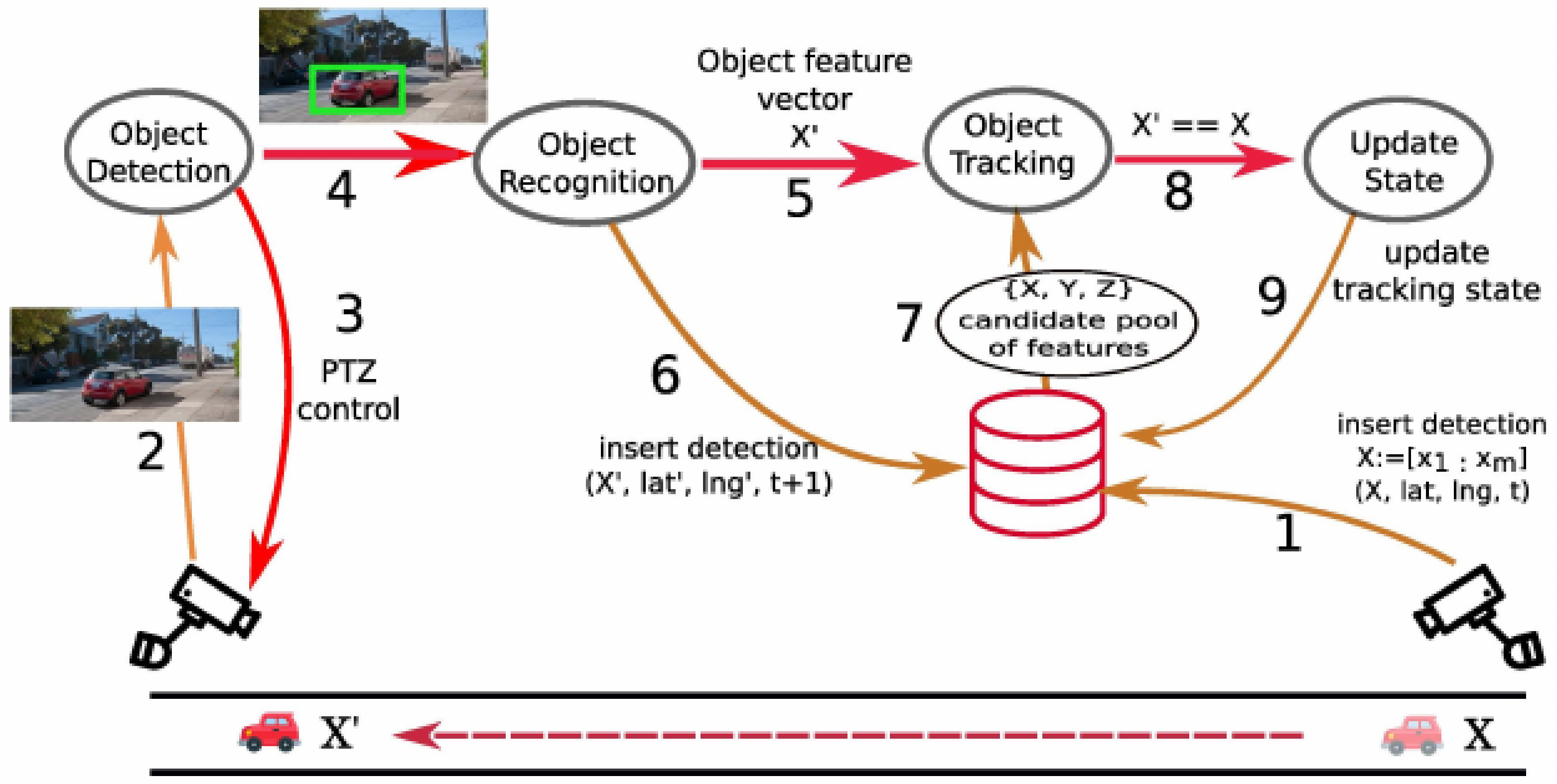

Exchange messages opportunistically
OK : +1-494-552-9433
OK : +1-494-552-9433


OK : +1-494-552-9433
OK : +1-494-552-9433



Use case: Suspicious vehicle tracking

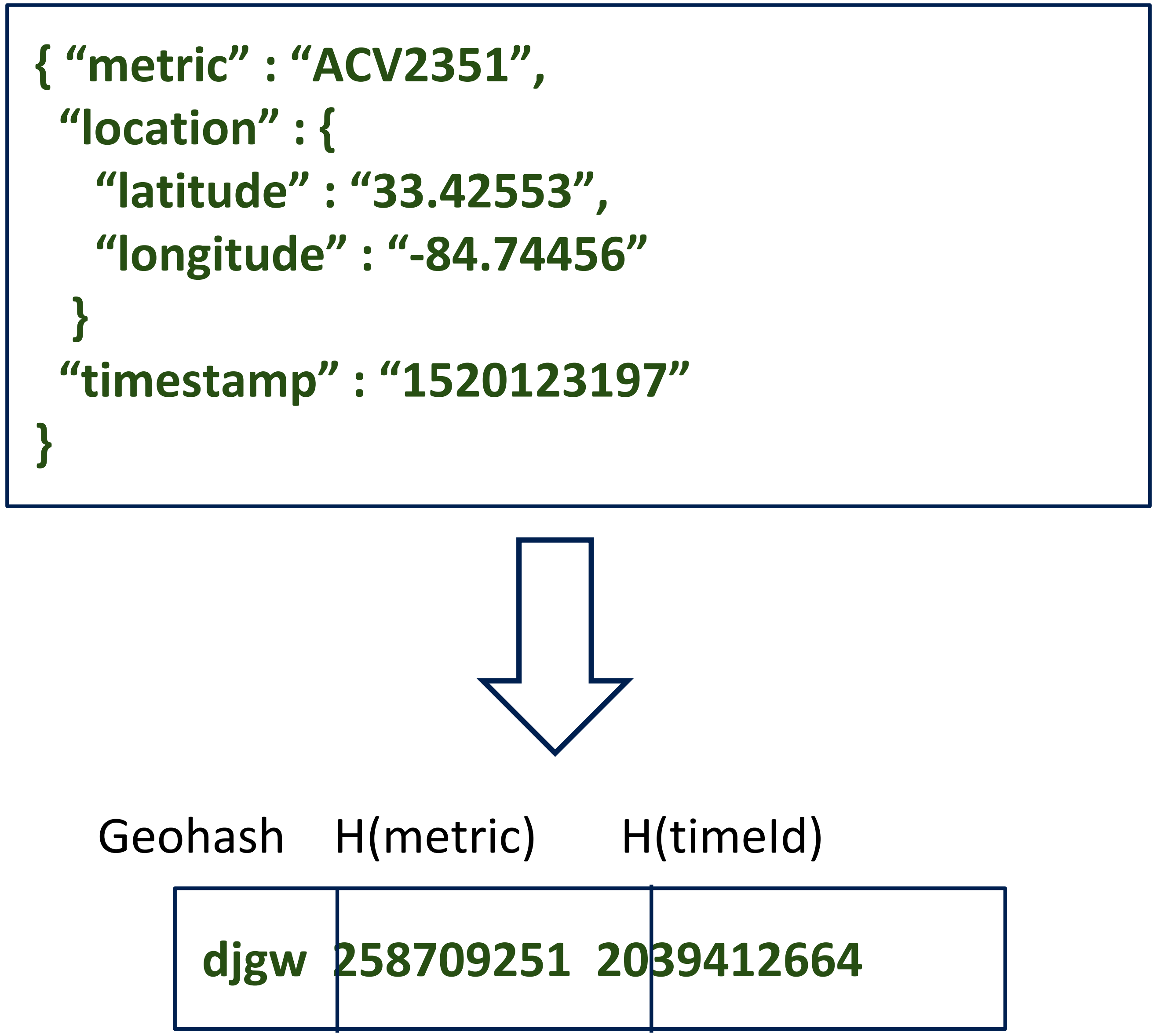
- Spatio-temporal range queries such as select all vehicle detections within 5km and 10 minutes to be efficient
- The distribution of workload is dependent on the distribution of vehicles in space, leading to hotspots
- For continuous operation, continuous streams of vehicle detections have to be saved in a datastore



DataFog

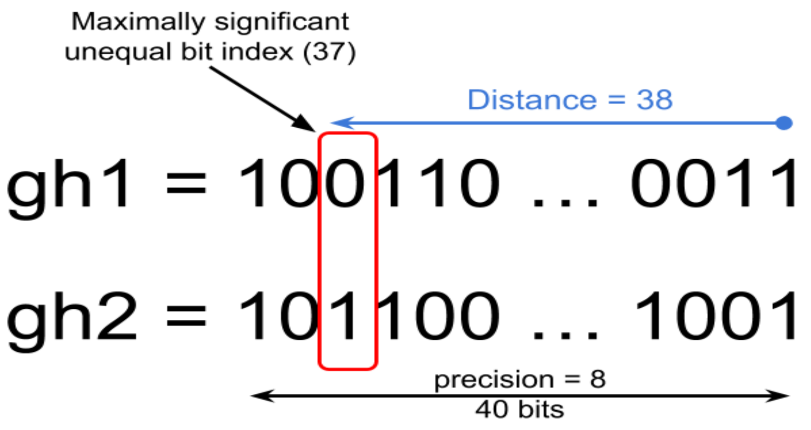
Locality-aware distributed indexing

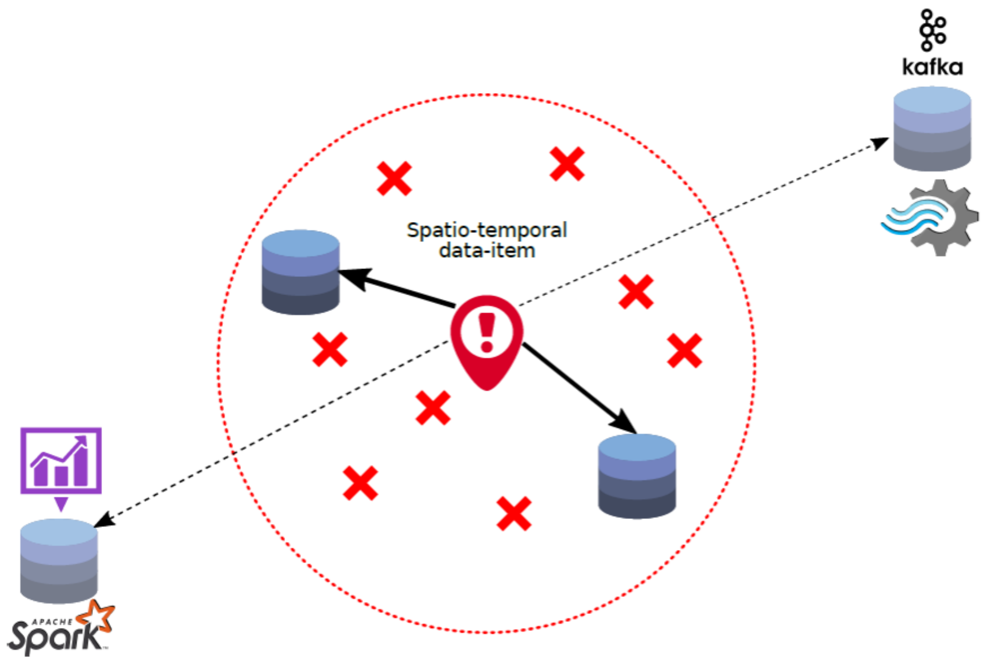
- Data-items are indexed based on their spatio-temporal attributes (e.g. Geohash)
- Consistent hashing for the location, timestamp and item-type attributes is used for partitioning data across nodes



Replication Policy

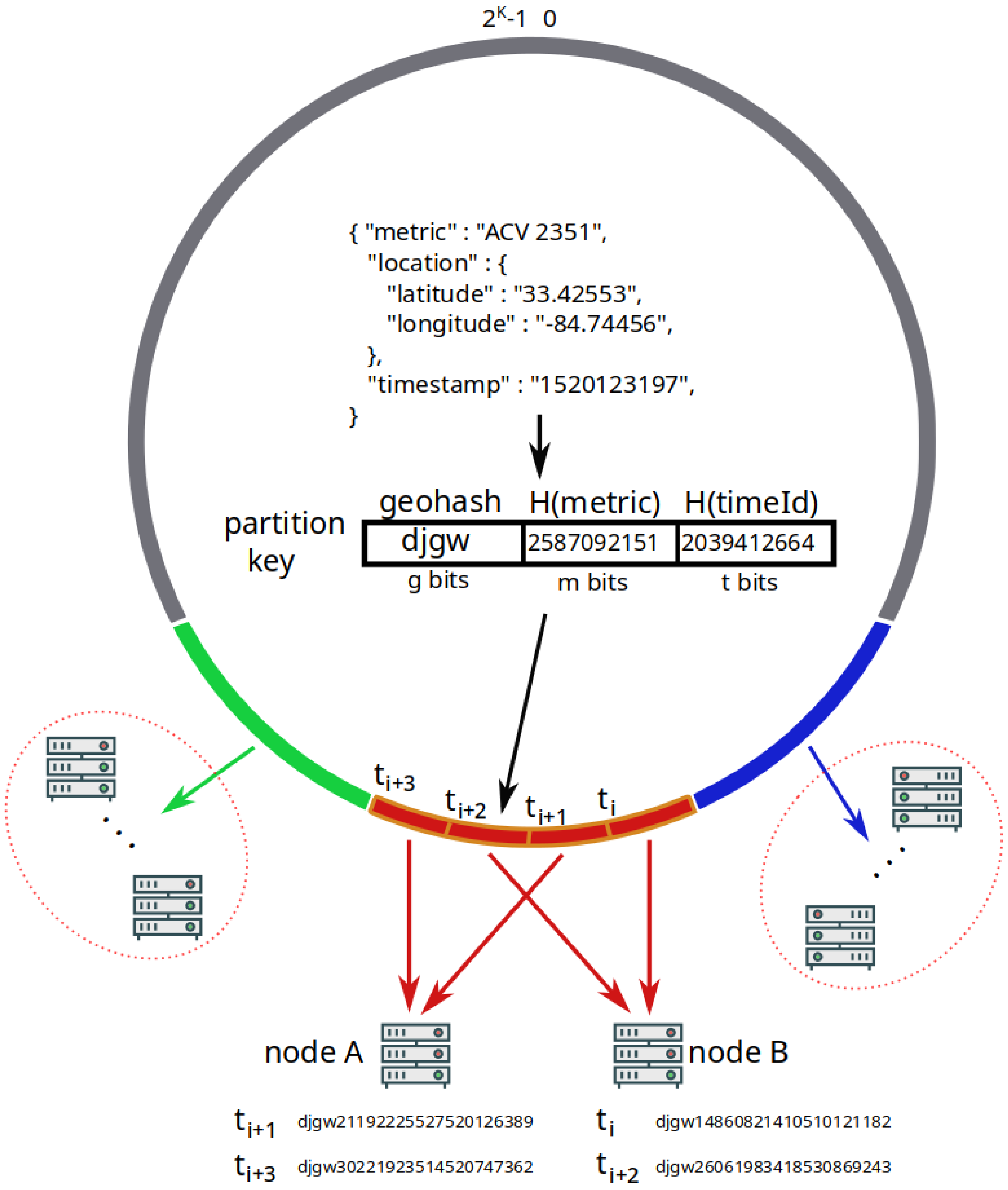
- Load-balancing and fault-tolerance
- Multiple replicas on Edge nodes for low latency
- Multiple replicas on remote datacenter nodes for tolerance from geographically correlated failures





Handling workload skews

- Load-balancing region
- Partition key -> virtual node -> physical node
- Mechanisms for adapting to hotspots
    - Long-lived: launch and attach new datastore nodes to the running cluster
    - Short-lived: offload heavily loaded nodes’ data items to lightly loaded nodes



Handling scarce resources at the edge

- TTL-based data eviction
  - Real-time analytics on temporal data
  - Batch-processing requires data spanning over a large period of time
- Data aggregation and compression
  - Omit redundant metadata to increase efficiency of storage utilization
  - Isomorphism of time series data

Non-closed Region and Boundary Cameras

- Create virtual cameras to connect all boundary cameras to force a closed region
- No theoretical activity upper bound for these virtual cameras
- However, in reality, vehicles active in a specified geographic region are largely "return" customers
- Archive trajectories from the virtual cameras into the cloud

Thank you

Microsoft