Microsoft

Research Faculty Summit 2018

Systems | Fueling future disruptions

Network Decomposition

Mung Chiang

Dean, College of Engineering, Purdue University

Founding Director, EDGE Lab, Princeton University

@OpenFog #fogcomputing

Network Decomposition

- Vertical & Horizontal Decomposition:
  - Who does what, at what timescale, how to glue them together?
  - Allocation of functions, not just resources
  - Vertical: Layering as decomposition
  - Horizontal: Cloud – Edge/Fog as decomposition

- Architecture supports Applications:
  - Source-channel separation: Digital communication
  - TCP/IP: Internet applications
  - Edge/Fog: IoT / 5G / Dispersive AI

OpenFog

The Princeton **EDGE** Lab

2009

Distribute functions to network edge

OpenFog

2015

Decompose functions along Cloud-2-Things Continuum

Interfaces

Massive storage
Heavy duty computation
Global coordination
Wide-area connectivity

Real time processing
Rapid innovation
User-centric
Edge resource pooling

OpenFog

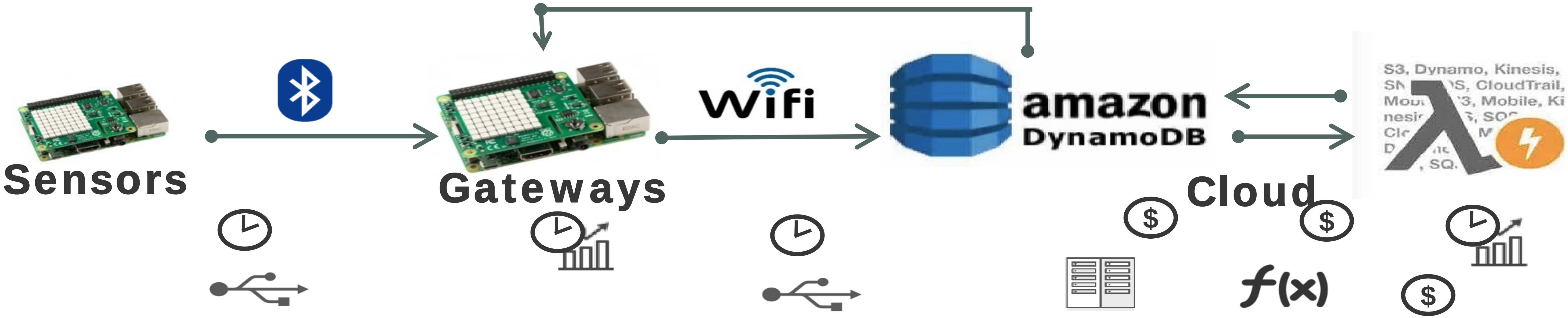
Applications to Dispersive AI

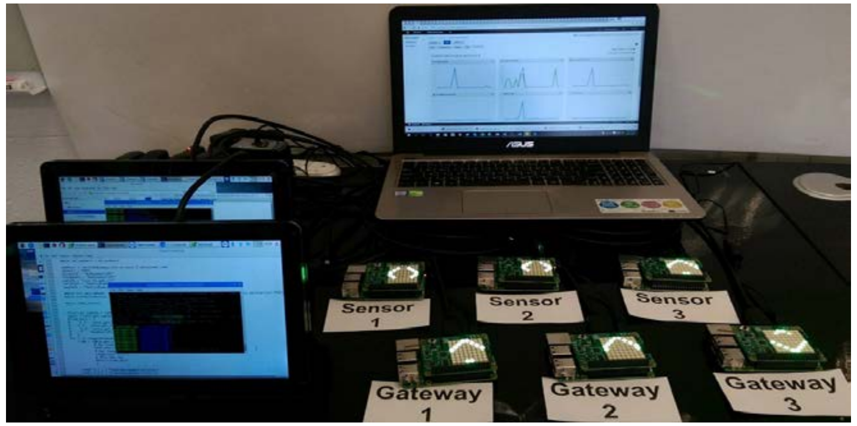
- Design machine learning algorithms that support fast responses
  - Enable IoT systems with intelligence here and now

- Decompose machine learning into multiple geographically distributed components, collectively or jointly operating
  - Optimize communication costs and centralized data processing costs
  - Make best use of local/proximal resources

- Proactively pre-position content and computing
  - Parallel successive refinement for streaming mining
  - Reduce infrastructure costs and improve quality of experience

Testing Decompositions of Machine Learning



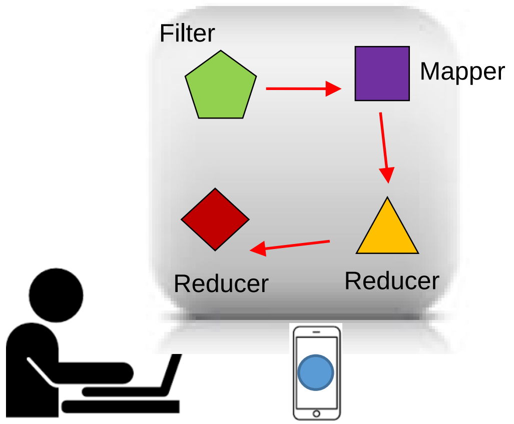


- Specialized testbed developed
- Linear regression decomposition: 70% reduction in data transmission
- Demonstrations: IEEE FWC’17, ACM SenSys’17 NYC Media Lab Summit’17.

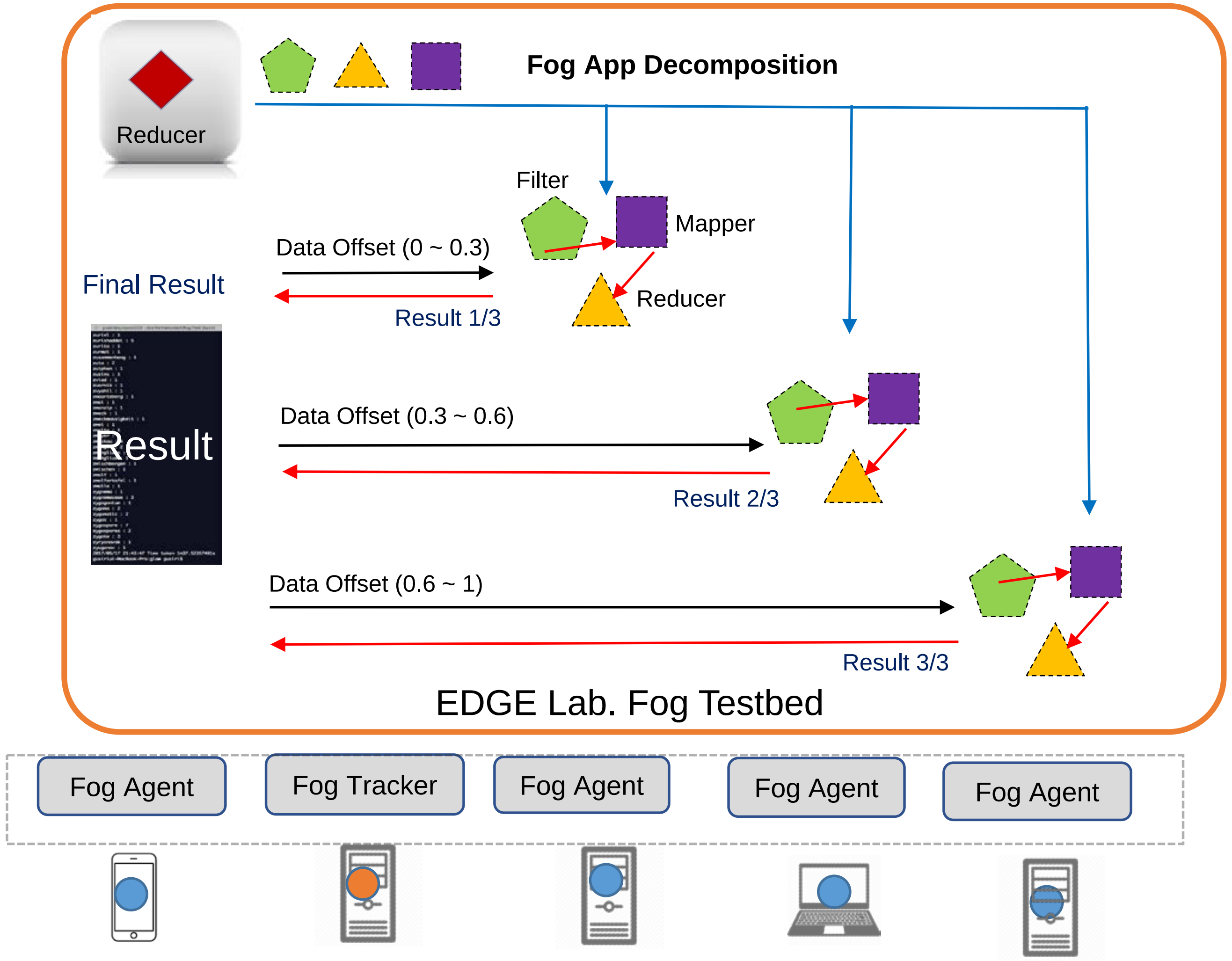
New MapReduce

- Re-visit MapReduce for Fog environment
  - Coordination & consistency
  - Robustness through redundancy
  - Computation decomposition
  - Task placement and scheduling along C2T
  - Automating deployment
- Prototype provides primary MapReduce APIs
  - Filter
  - Map
  - Reduce
  - ReduceByKey
  - MapByKey

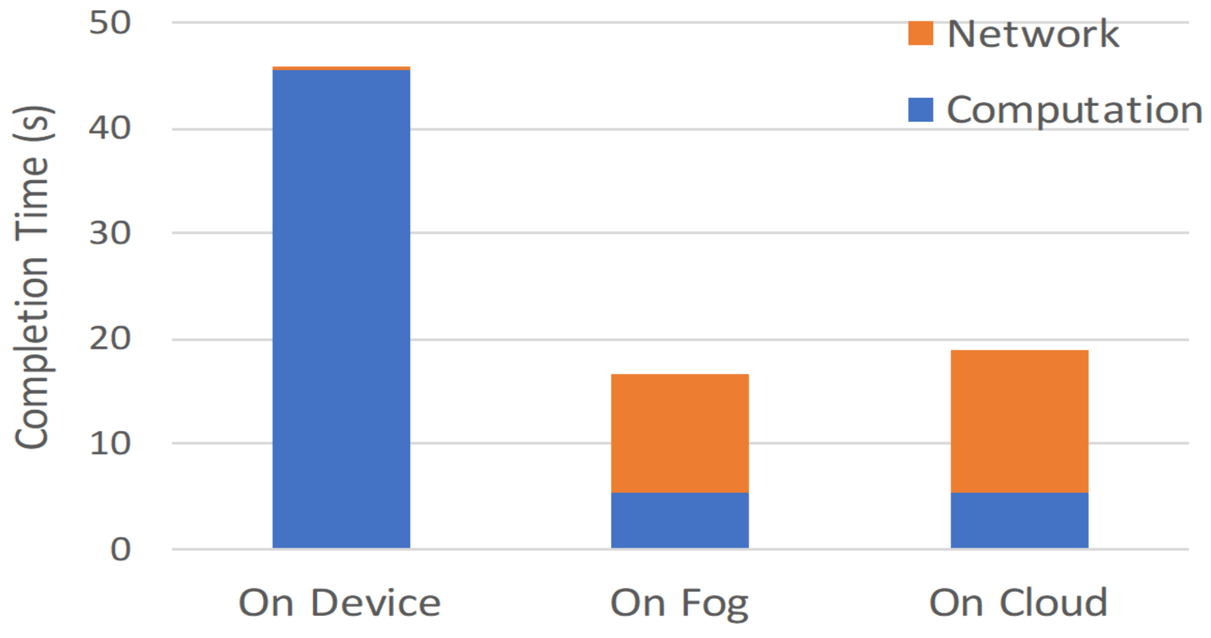
Local Fog MapReduce

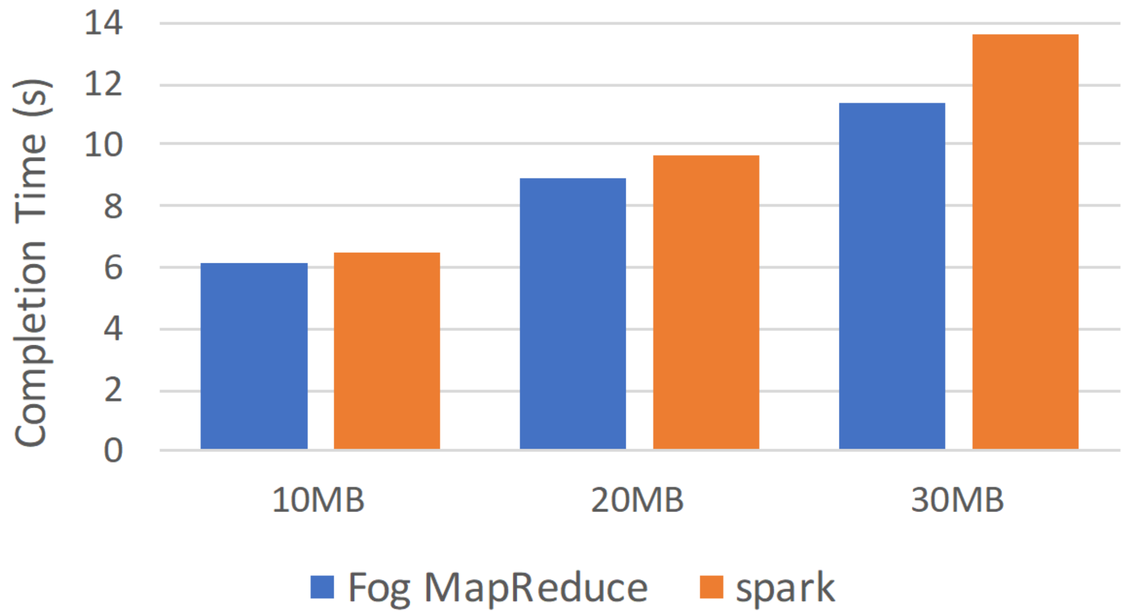


Distributed Fog MapReduce



Simple Example: Counting Words in a File





Unique Features in Edge/Fog

- Heterogeneity/Under-organization of resources/devices
- Variability/Volatility in availability/mobility
- Bandwidth/Battery constraints
- Proximity to sensors/actuators

OpenFog

Two Toy Examples

- Drone Camera Network
  - Aakanksha Chowdhery, Shirley Wang

- Personalized Placement Learning in AR/VR
  - Surin Ahn, Maria Gorlatova, Parinaz Naghizadeh

Outdoor Live Coverage

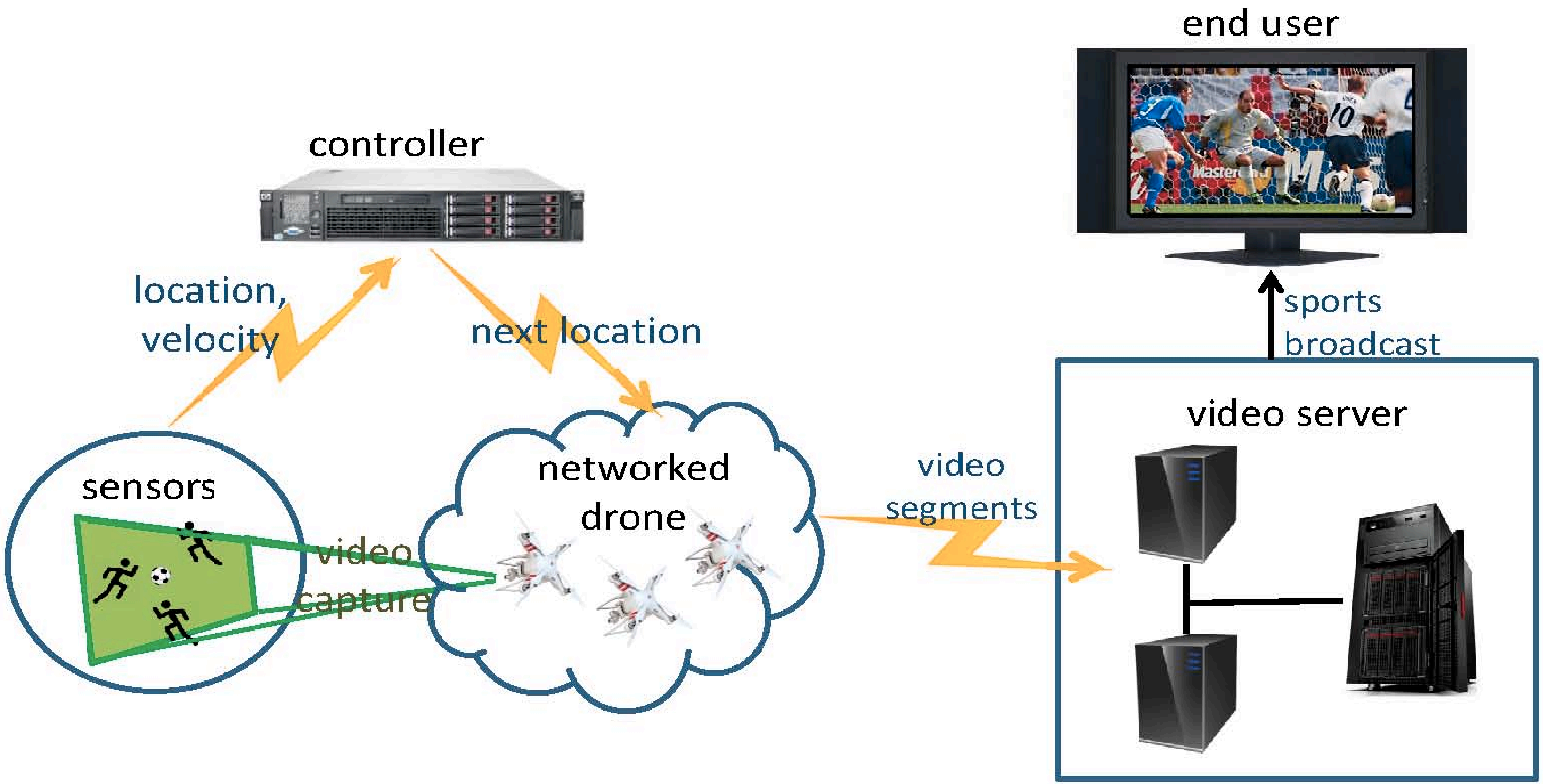
- Stadiums use 30-50 cameras to capture games nationally broadcast

- NFL recently introduced cameras over crosswires: SkyCam, SpiderCam

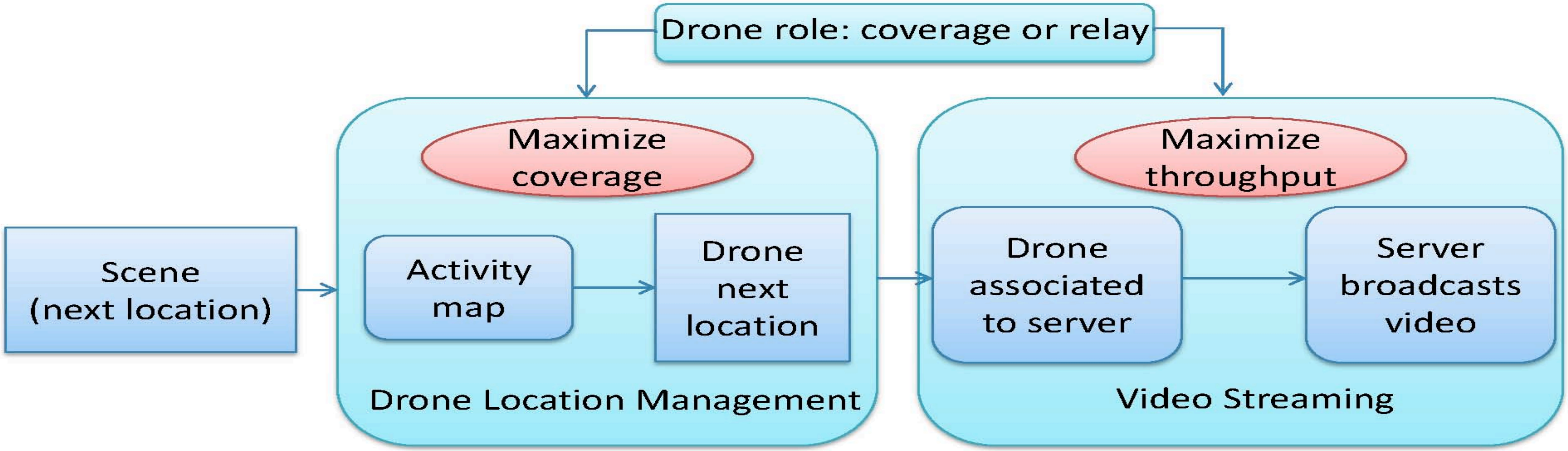
Network Decomposition

- Distributed
  - Make decision locally; low latency
  - Suboptimal: all drones prioritize covering the most important scene
- Centralized
  - Cloud makes decisions, cost benefits and efficiency
  - Round-trip time limits drone trajectory coordination rate, response to game dynamics
- One decomposition:
  - Drones form a fog network: trajectory planning, collision avoidance
  - Centralized coordination for next target location, drone function

One Choice of Decomposition



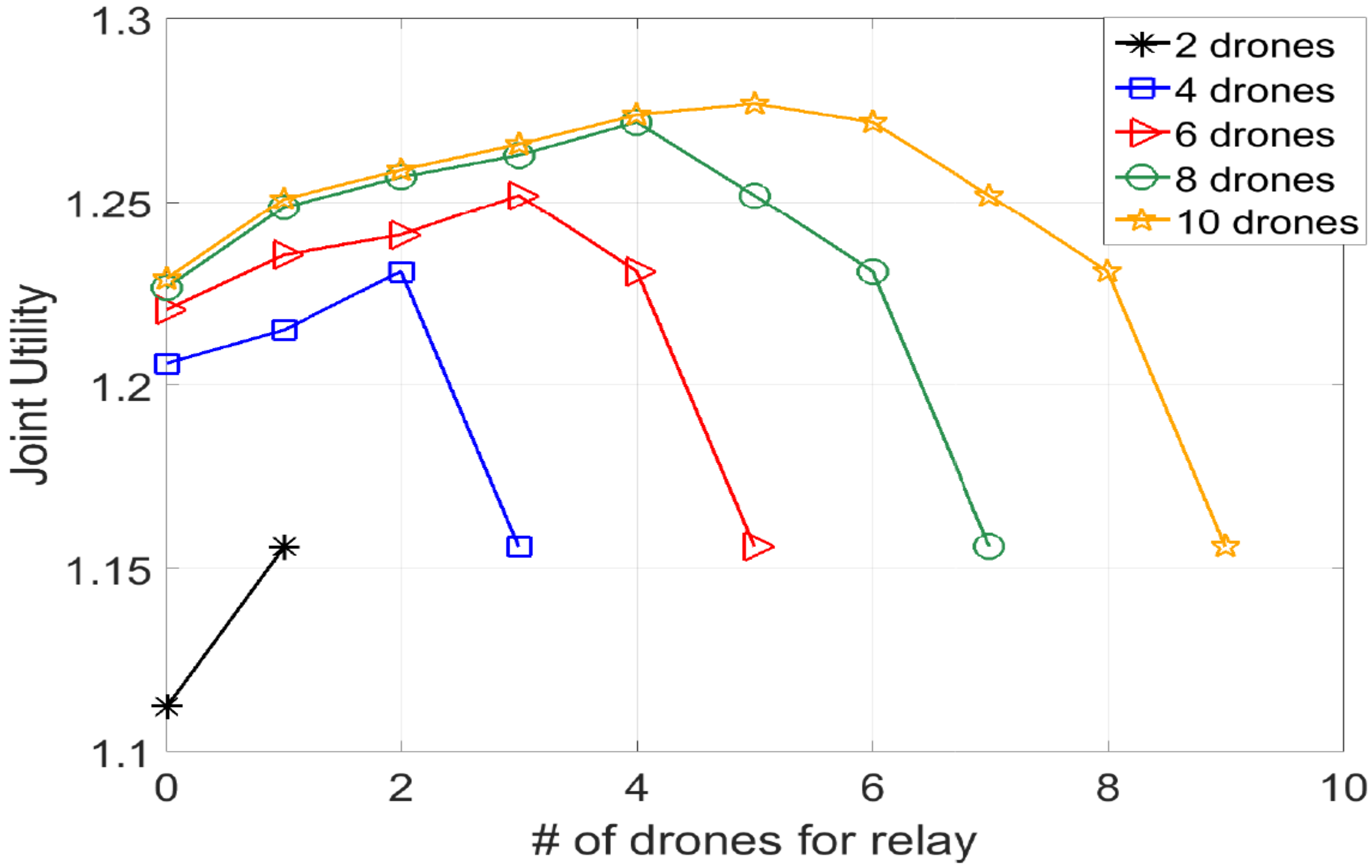
Coverage vs. Throughput



- Sensors on players/ball (location, speed)
- Predict locations of next importance scenes
- Match drones to each important scene

- Max throughput: drone selects the best server
- Throughput boosting: drone as relay

Joint Optimization and drone allocation



Optimal allocation of drones as relay

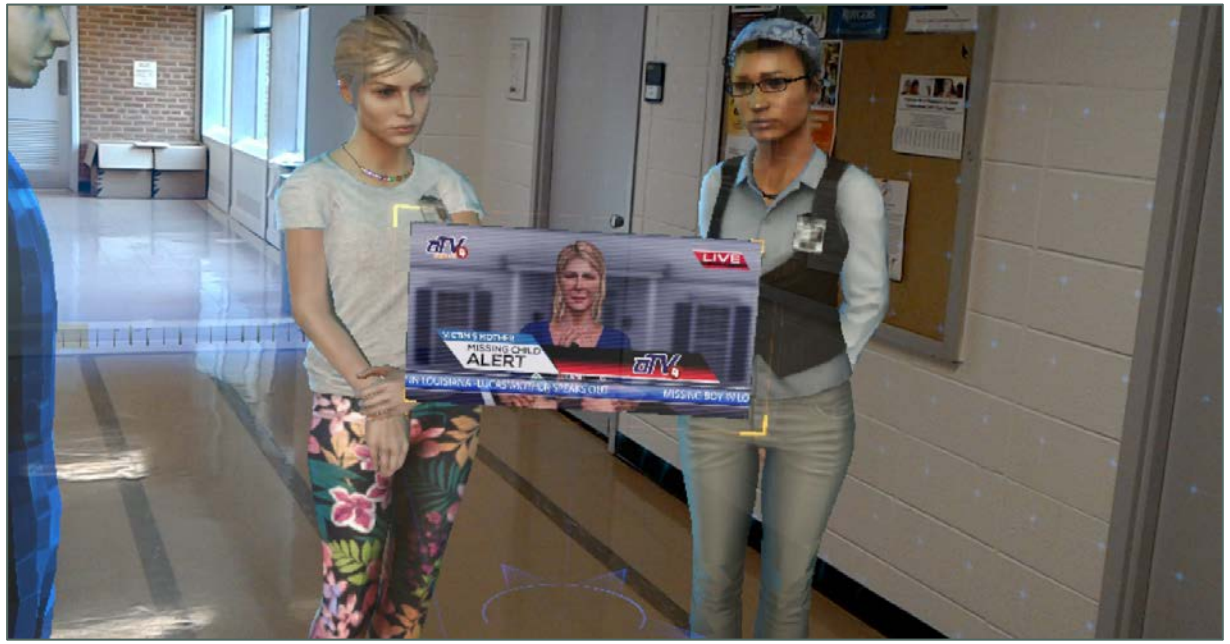
Observations

- A system to coordinate a network of drones to capture and broadcast a sports game from drone cameras over a wireless channel

- Joint Optimization
  - **Maximize the coverage of interesting scenes**: drone location assignment algorithm.
  - **Maximize video quality**: drone-server association scheme, drone as relay.

- Evaluation: optimal 4 drones for coverage, 4 as relay
  - achieve 94% coverage
  - support 2K video streaming at 20Mbps

OpenFog

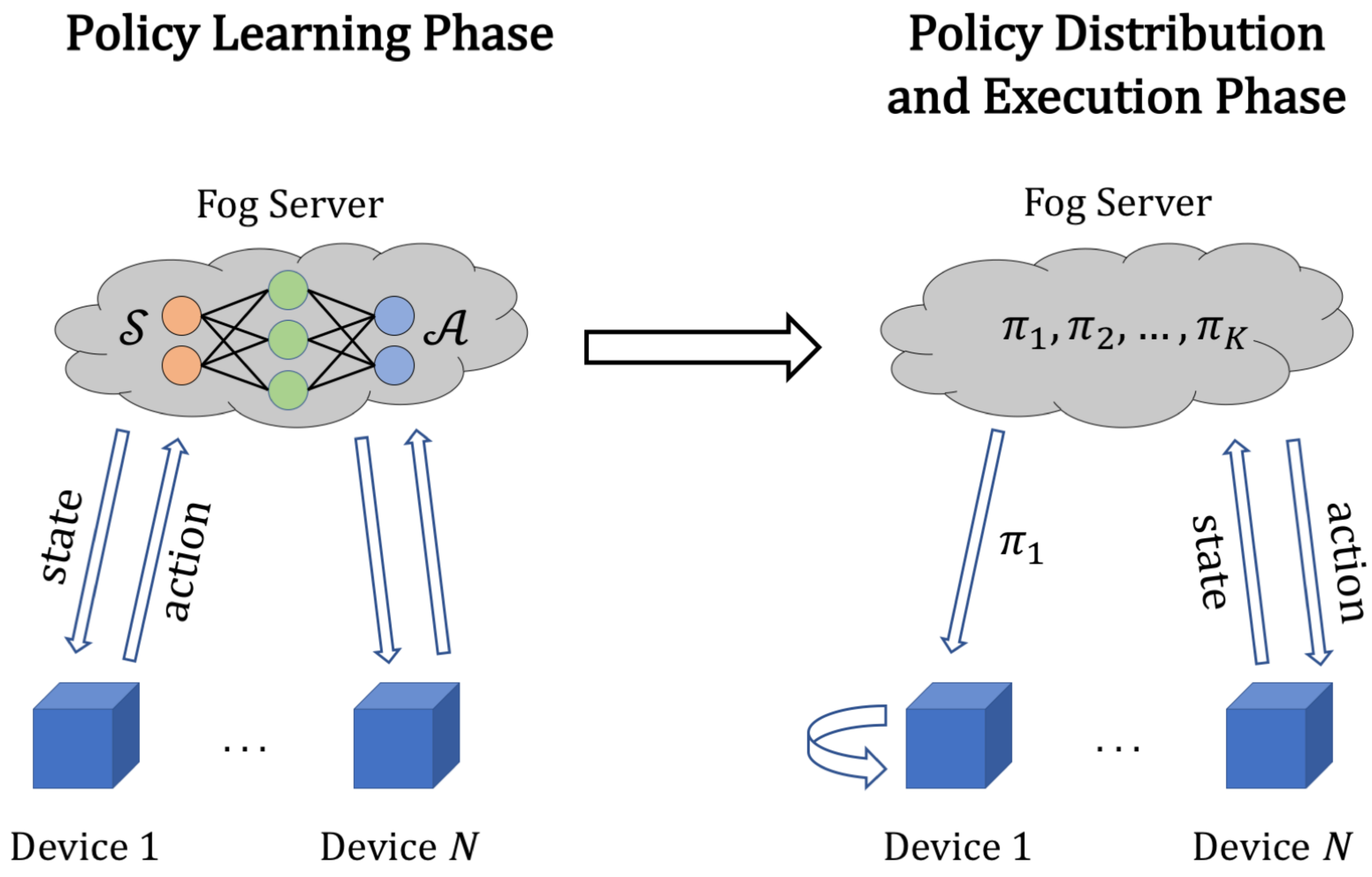


Personalized AR Object Placement:

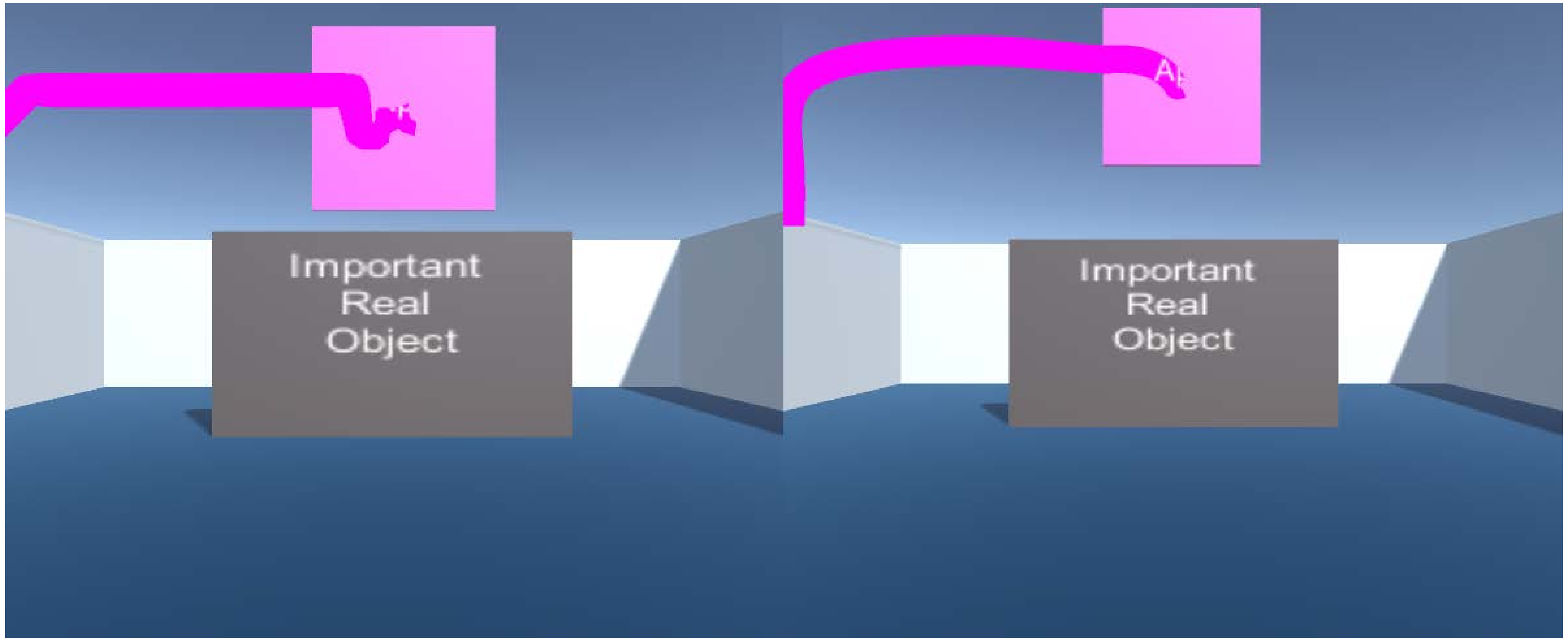
Where Does Learning Happen?

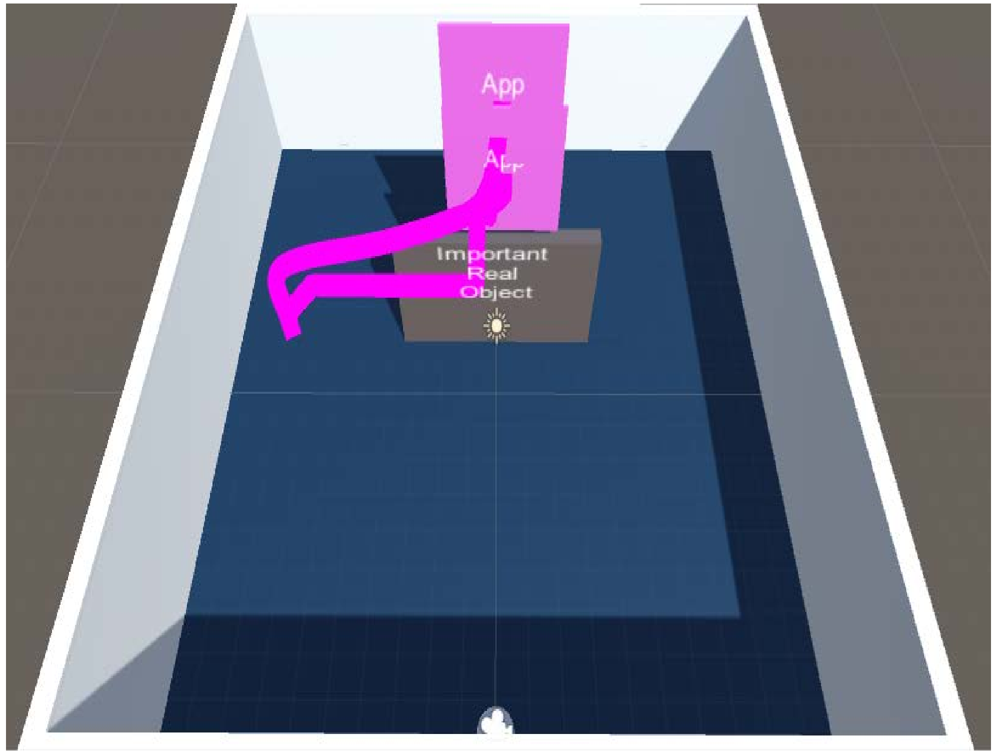
Network Decomposition

- **Policy learning phase:**
  - Present users with simulated environments; collect their actions
  - Use imitation learning to learn personalized hologram placements
- **Policy execution phase:**
  - Send to powerful devices
  - Run on behalf of resource-constrained devices



Personalized Placement Learning

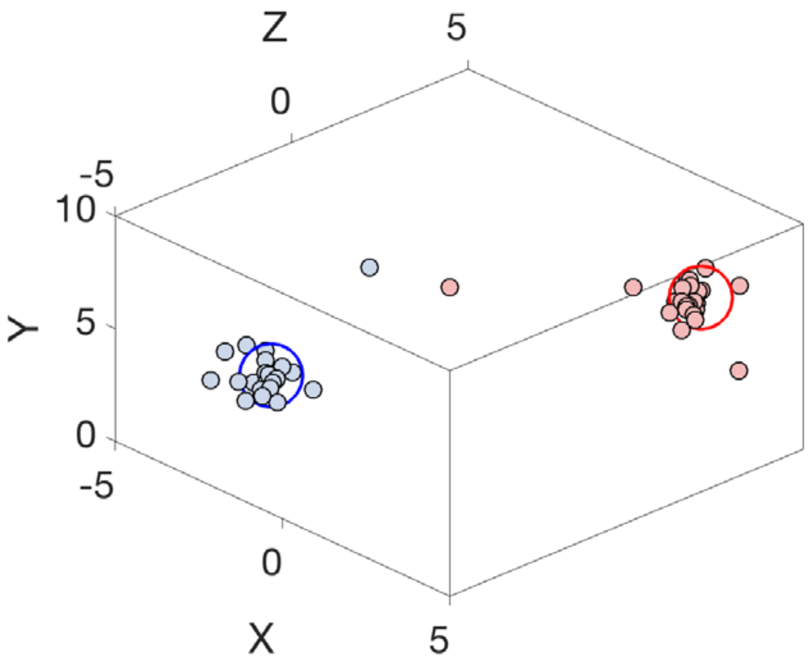




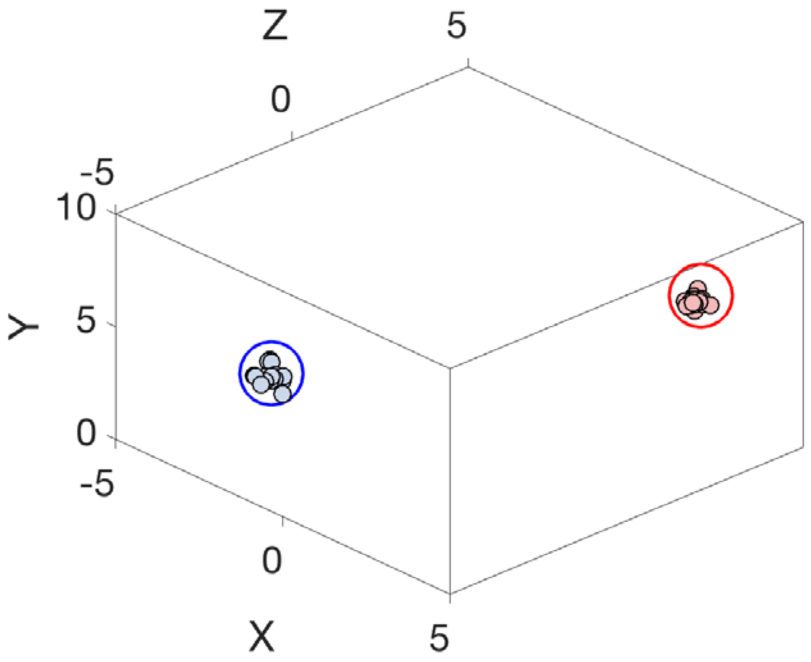
User-controlled hologram Agent-controlled hologram

Birds-eye view shows matching trajectories in addition to placements

Learning A User’s Preference Fast



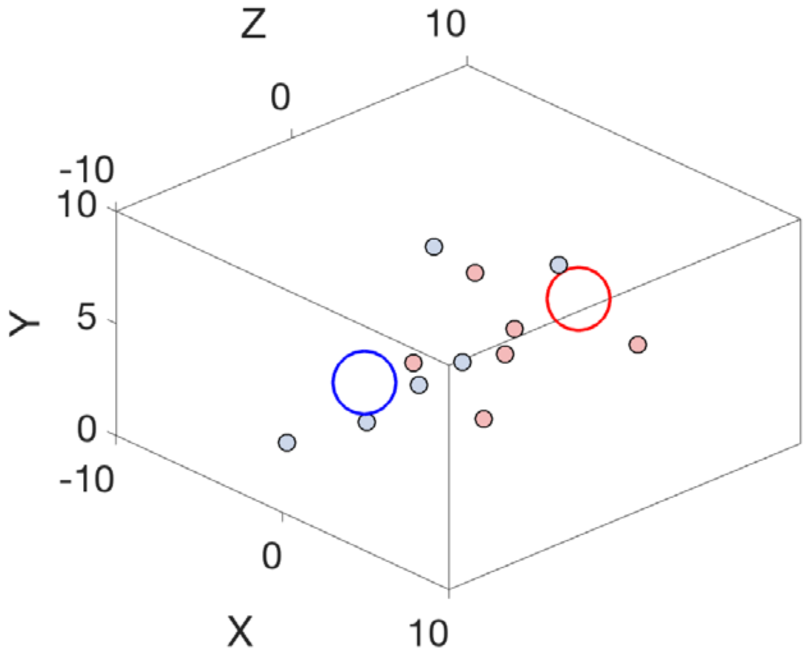
4
Demos



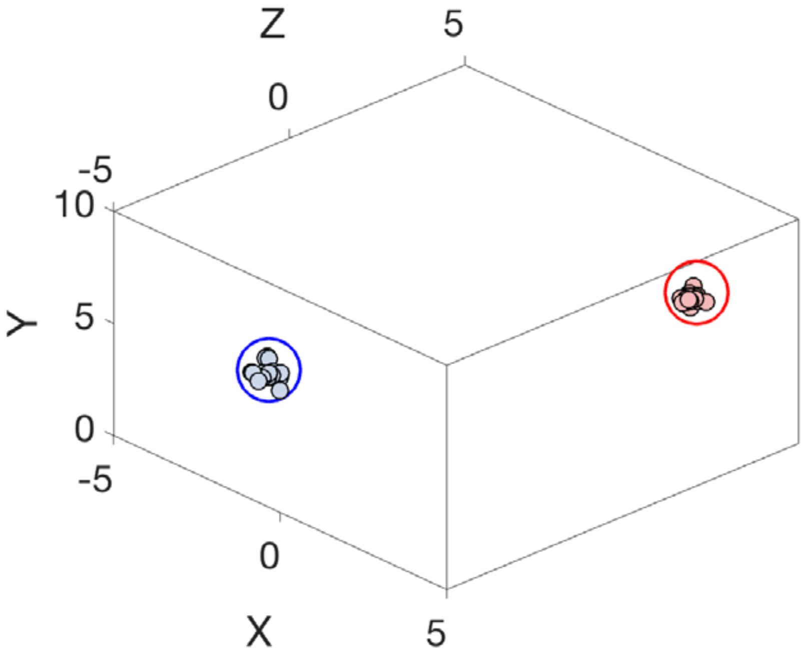
13
Demos

Each circle represents the users’ desired destination. Dots represent algorithm’s action. As early as 13 demos, agent achieves accuracy.

State Space Reduction Helps



Full image as state



Reduced state-space

Hologram destinations after 12 trials when using the full AR image as the state. Compared with the reduced state-space there is degradation in accuracy.

Overarching Questions

- Where to place a function/data set?
- What is the interface among modules?

The Edge/Fog Advantage: SCALE

- Security
- Cognition
- Agility
- Latency
- Efficiency

Thank you!

Microsoft