The Emergence of Edge Computing

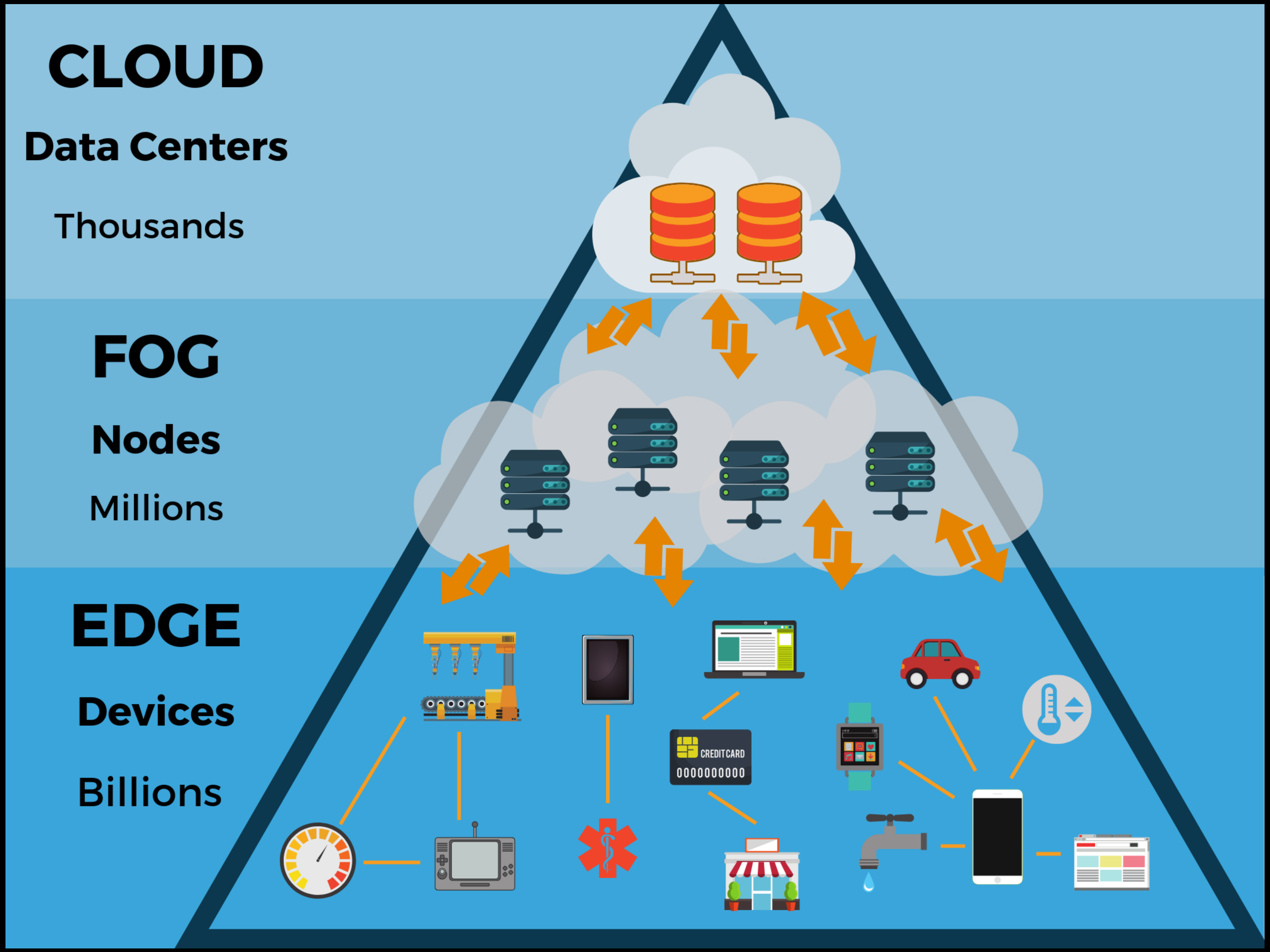
Satyanarayanan, Mahadev. "The Emergence of Edge Computing." Computer 50 (2017): 30-39.
Nanda Velugoti

Overview

- Origin and Background of Edge Computing
- What is a Cloudlet?
- Importance of Proximity
- How the proximity of cloudlets help (in four distinct ways)?
- Future of Edge Computing

Origin and Background

- Akamai introduced *content delivery networks* (CDNs), in late 1990s, to accelerate web performance.
- In 1997, Brian Noble showed how speech recognition could be implemented on a resource-limited (mobile) device by *offloading computation* to near by server.
- In mid 2000's, *cloud computing* came about and became popular.
- In 2009, Satya coauthored a paper that introduced the concept of *cloudlets*.
- In 2012, introduced the term *fog computing* to refer to the dispersed cloud infrastructure.



CDN vs Cloudlet

- A *CDN* uses nodes at the edge close to users to prefetch and cache web content.
- A *cloudlet* can run arbitrary code just as in cloud computing. This code is typically encapsulated in a virtual machine (VM) or a lighter-weight container for isolation, safety, resource management, and metering.
- In both cases, *proximity is crucial*.

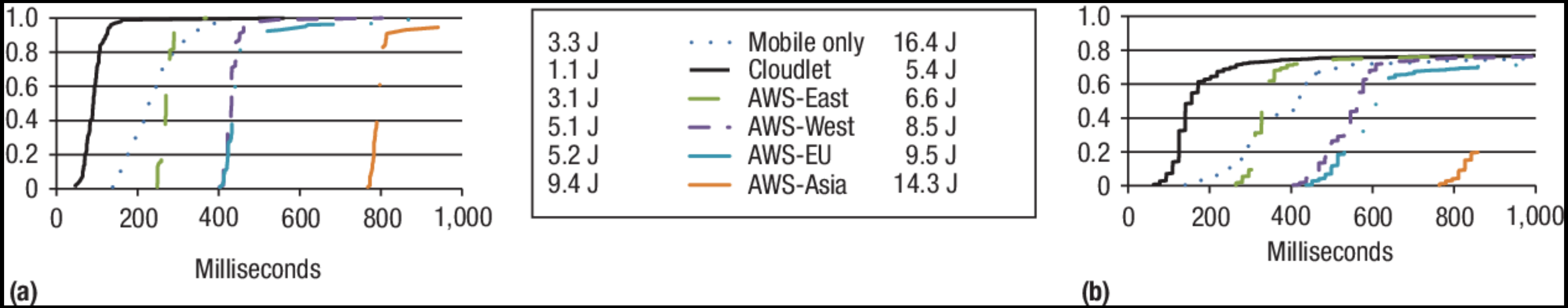
Importance of Proximity

- Ang Li and his colleagues reported that the average round-trip time from 260 global vantage points to their optimal AWS EC2 instances is *~74 ms*, excluding the latency of a wireless first hop.

- But emerging applications, such as AR/VR and cloud based gaming, require end-to-end delays to be tightly controlled to less than *~10 ms*.

- We will discuss how the proximity of cloudlets helps in at least four distinct ways.

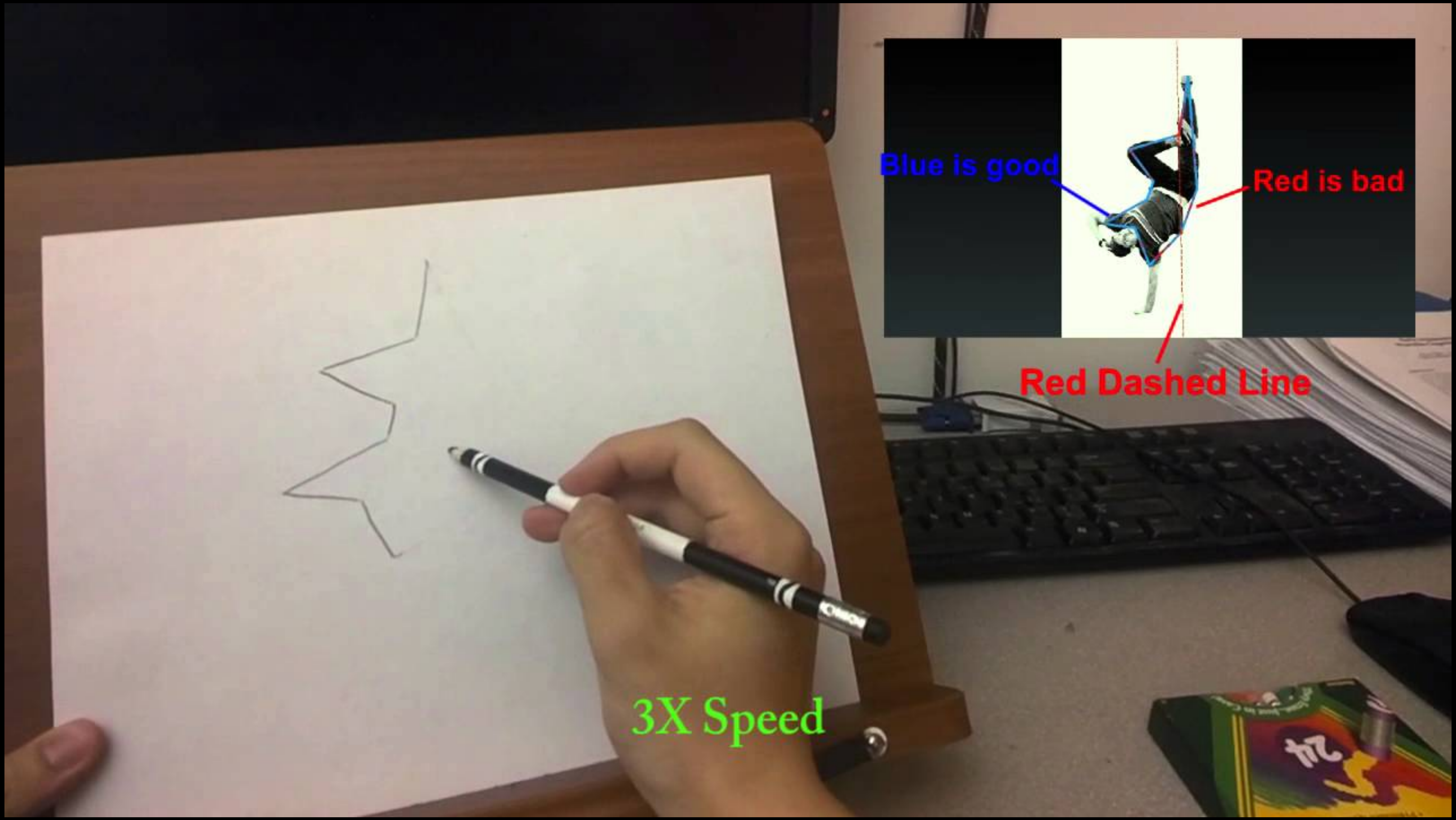
1. Highly Responsive Cloud Services



Response time distribution and per-operation energy cost of an (a) augmented reality and (b) face recognition application on a mobile device

- Humans are acutely sensitive to delays while interacting or performing cognitive tasks.
- Face recognition takes *370-620 ms*.
- Speech recognition takes *300-400 ms*, short phrases require only *4 ms*.
- VR applications that use head-tracked systems require latencies *< 16 ms*.

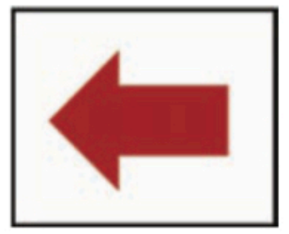
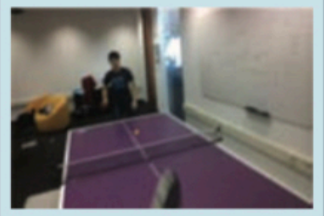
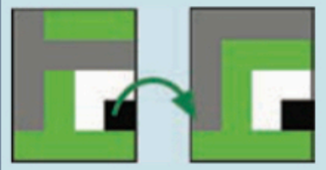
Drawing Assistant with Google Glass



Cloudlet Workflow

- *Phase 1*: The sensor inputs are analyzed to extract a symbolic representation of task progress (4th column in table).
- *Phase 2*: Compares the symbolic representation to the expected task state generates user guidance for the next step (last column in table).

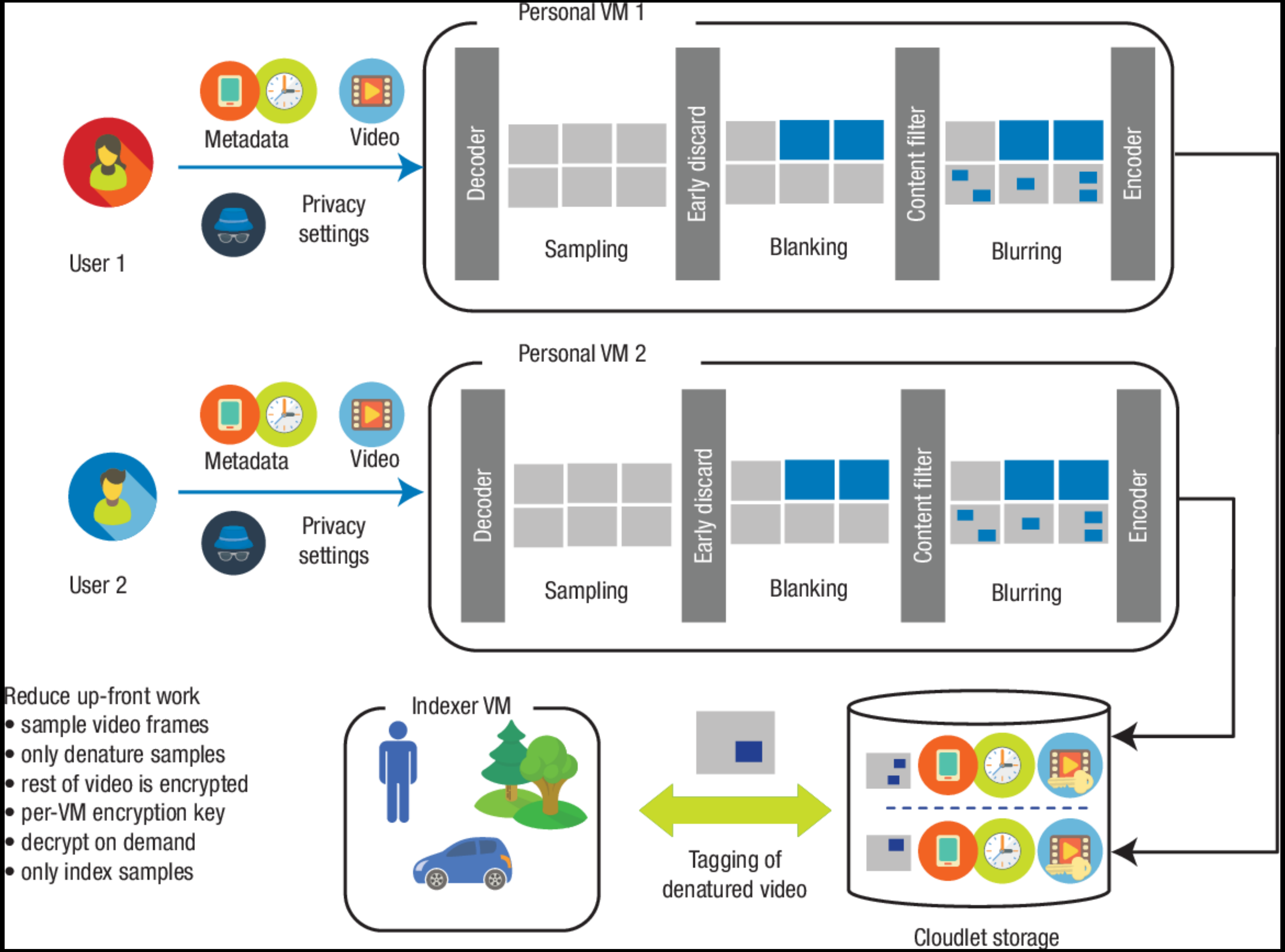
TABLE 1. Example wearable cognitive assistance applications.

App name	Example input video frame	App description	Symbolic representation	Example guidance
Face		Jogs user's memory of a familiar face whose name cannot be recalled. Detects and extracts a tightly cropped image of each face, then applies popular open source face recognizer OpenFace (cmusatyalab.github.io/openface), which is based on a deep neural network (DNN) algorithm. Whispers name of person. Can be used in combination with mood detection algorithms to offer conversational hints.	ASCII text of name	Whispers "Barack Obama"
Pool		Helps novice pool player aim correctly. Gives continuous visual feedback (left arrow, right arrow, or thumbs up) as user turns cue stick. Correct shot angle is calculated based on widely used fractional aiming system. Uses color, line, contour, and shape detection. Symbolic representation describes positions of cue ball, object ball, target pocket, and top and bottom of cue stick.	<Pocket, object ball, cue ball, cue top, cue bottom>	
Ping-Pong		Tells novice player to hit ball to left or right, depending on which is more likely to beat opponent. Uses color, line, and optical-flow-based motion detection to detect ball, table, and opponent. Symbolic representation is a 3-tuple: in rally or not, opponent position, ball position. Whispers "left" or "right."	<InRally, ball position, opponent position>	Whispers "Left"
Workout		Guides correct user form in exercise actions like sit-ups and push-ups, and counts out repetitions. Uses volumetric template matching on a 10- to 15-frame video segment to classify poorly performed repetitions as distinct types of exercise (for example, "bad push-up"). Uses smartphone on floor for third-person viewpoint.	<Action count>	Says "8"
Lego		Guides user in assembling 2D Lego models. Analyzes each video frame in three steps: (1) finds board using its distinctive color and black dot pattern, (2) locates Lego bricks on board using edge and color detection, and (3) assigns brick color using weighted majority voting within each block. Symbolic representation is matrix showing color for each brick.	[[0, 2, 1, 1], [0, 2, 1, 6], [2, 2, 2, 2]]	 Says "Find a 1 × 3 green piece and put it on top"
Draw		Helps user to sketch better. Builds on third-party app originally designed to input sketches from pen tablets and output corrective guidance on desktop screen. Our implementation preserves back-end logic. New Google Glass–based front end allows use of any drawing surface and instrument and displays guidance on Glass. Displays error alignment in sketch.		
Sandwich		Helps cooking novice prepare sandwiches according to a recipe. Because real food is perishable, we use realistic plastic toy food as ingredients. Object detection uses a region proposal and DNN approach. Implementation is on	Object: "Lettuce on top of ham and bread"	

2. Scalability via Edge Analytics

- Cloudlets can also *reduce* ingress bandwidth into the cloud.
- Problem: The cumulative data in a modest city of 12,000 users transmitting 1080p video would require a link of *100 gigabits per second*; a million users would require a link of *8.5 terabits per second*.
- Solution: The cloudlet runs computer vision analytics in near real time and only sends the results (content tags, recognized faces, and so on) along with metadata (owner, capture location, timestamp, and so on) to the cloud. Thus reducing the ingress bandwidth.

GigaShight Framework



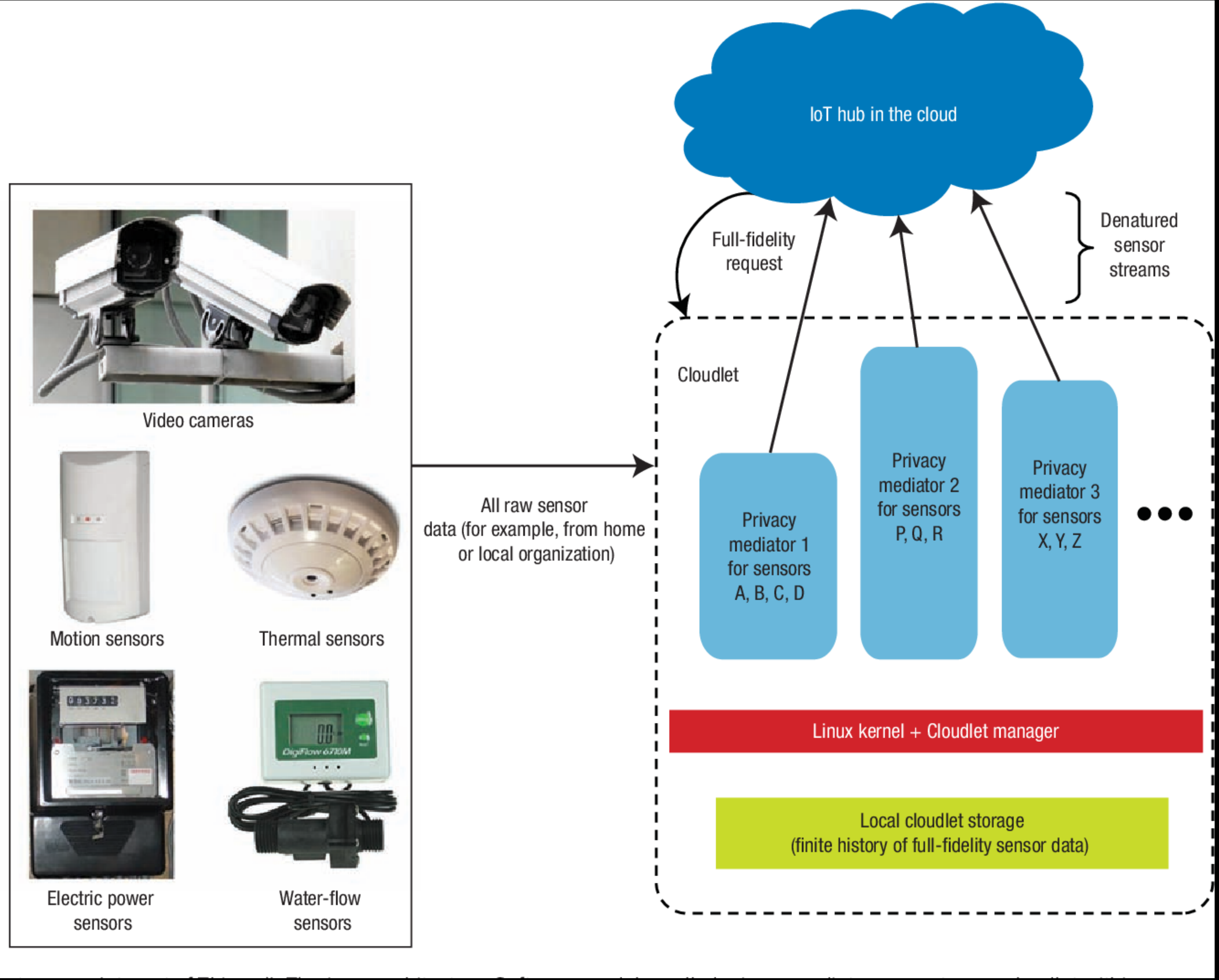
3. Privacy-policy Enforcement

- From the end user's perspective, *denatured sensor data* is safe to release to the outside world: faces in images can be blurred, sensor readings can be coarsely aggregated or omitted at certain times of day or night, and so on.

- Problem: Today's IoT architectures, in which data is transmitted directly from sensors to a cloud hub, make such fine-grain control impossible.

- Solution: Trusted software modules called *privacy mediators* execute on the cloudlet to perform denaturing and privacy-policy enforcement on the sensor streams.

IoT Privacy Architecture



4. Masking Cloud Outages

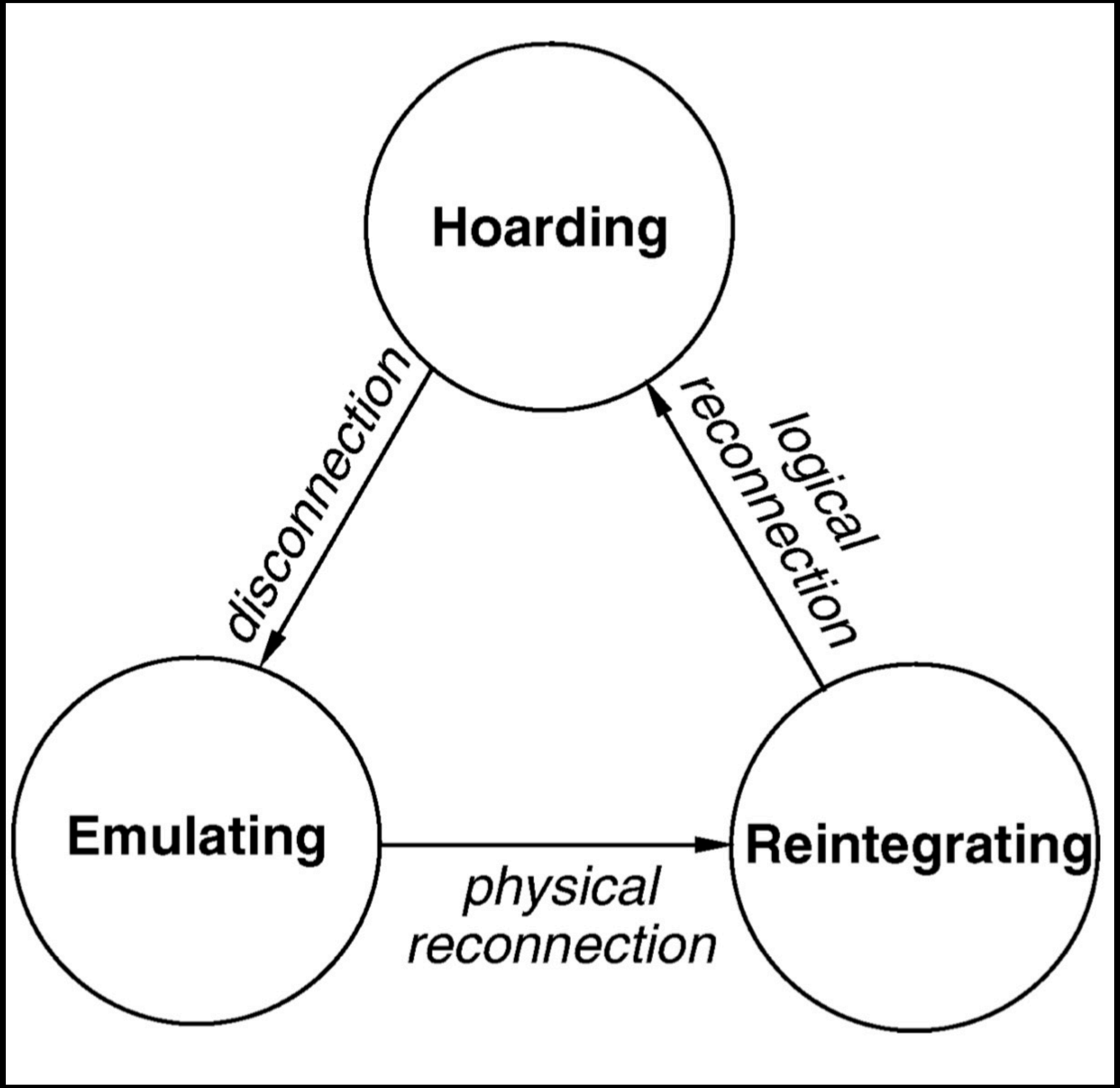
- *Hostile environments* is a viewpoint that applies to several important contexts where access to cloud must be viewed as an occasional luxury rather than a basic necessity.

- Examples: Military operations, complex geographical regions, cyber attacks, etc.

- This opens the door to approaches in which a fallback service on a cloudlet can temporarily *mask* (by acting as a proxy for) cloud inaccessibility.

Coda File System

- Coda File System provides disconnectable read–write access to shared data.
- *Hoarding* - prefetching data into a persistent cache
- *Emulation* - leveraging hoarded data in the cloud's absence and precisely tracking local updates
- *Reintegration* - propagating updates to the cloud, and detecting and resolving conflicts



What’s next?

- Generalizing steps in Coda File System to various cloud services will be an important future research area
- How to manage dispersed edge nodes? How to ensure security?
- What is the incentive to use edge computing for business?
- Deadlock: infrastructure vs applications. How can we break it?
- Centralized vs decentralized computing dialectic
- Steps in right direction: OpenStack++, SDN, NFV, 5G, IoT