Gabriel: Towards Wearable Cognitive Assistance

Kiryong Ha, **Zhuo Chen**, Wenlu Hu, Wolfgang Richter,
Padmanabhan Pillai, and Mahadev Satyanarayanan

Carnegie Mellon University

intel

Cognitive Decline

survivors of stroke

mild cognitive impairment

One-month delay in nursing home admissions saves **$1,000,000,000+/year**

20,000,000+ Americans affected

Alzheimer's disease

traumatic brain injury

Faces

Text

Daily Routine

Can Wearable Technology Help?

Continuously capture, interpret, and give guidance

System Architecture

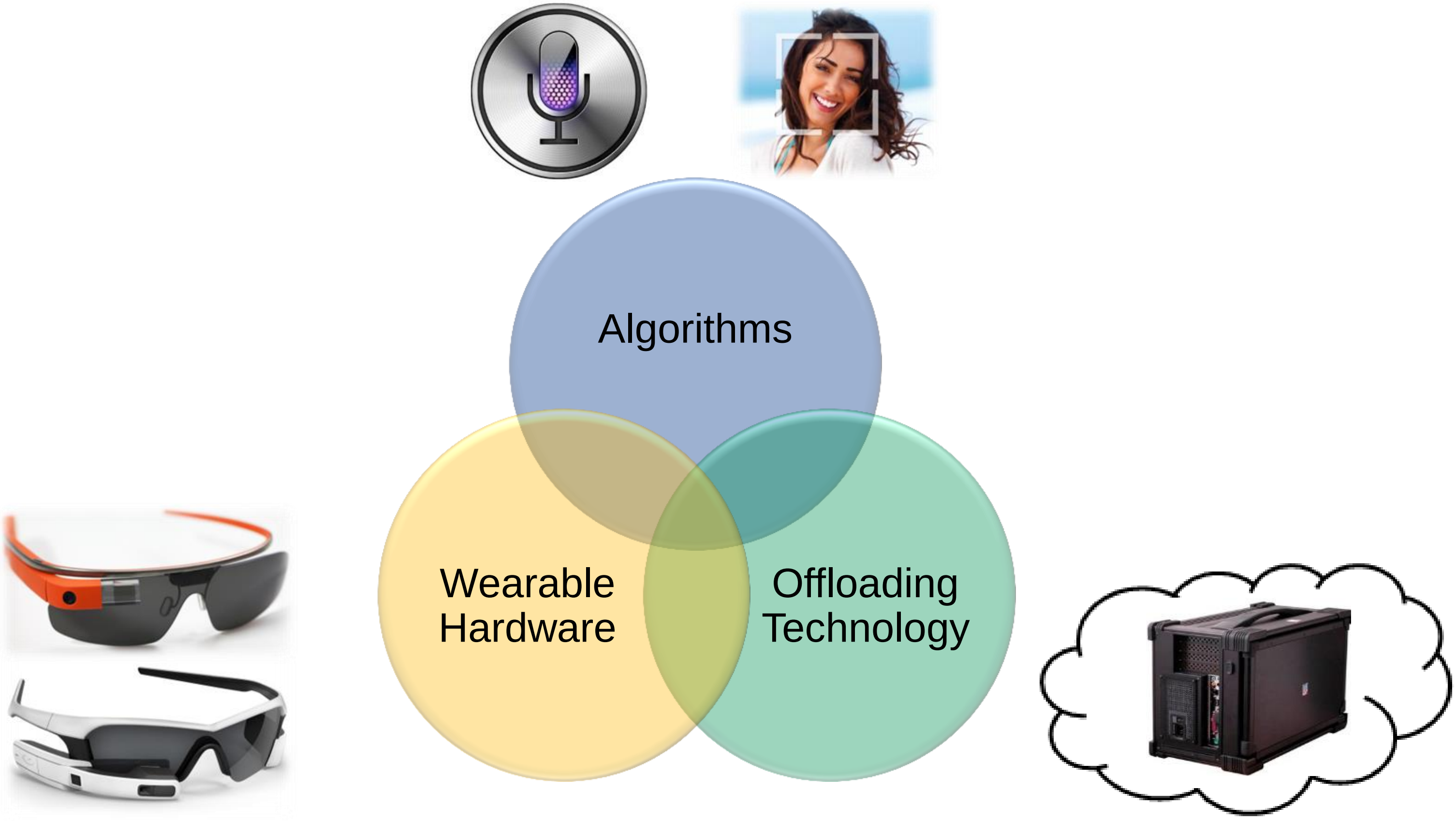
“Barack is saying hello to you”

“Please stop and check traffic”

“Your dog wants to go out for a walk”

Why Today?

Advances in 3 Independent Arenas



Algorithms

Wearable Hardware

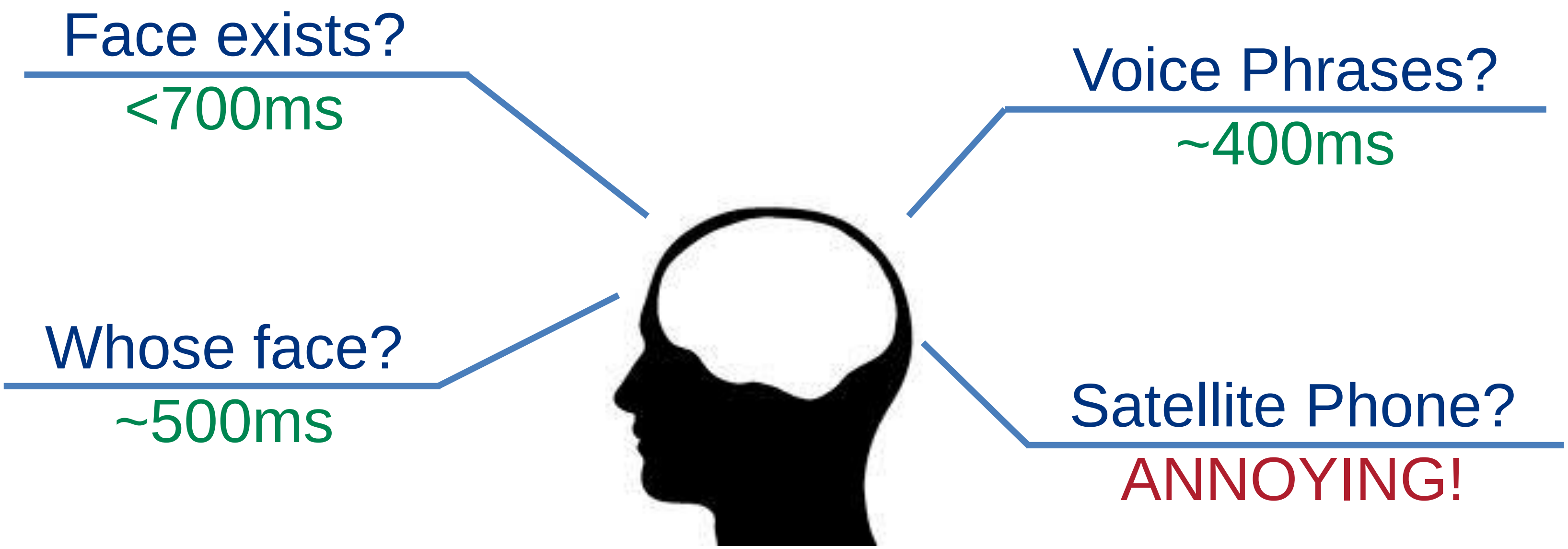
Offloading Technology

Challenges – Architecture

1. Crisp Interactive Response
2. Graceful Degradation of Services
3. Coarse-grain Parallelism

C1. Crisp Interactive Response

Humans are fast and sensitive



Face exists?

<700ms

Voice Phrases?

~400ms

Whose face?

~500ms

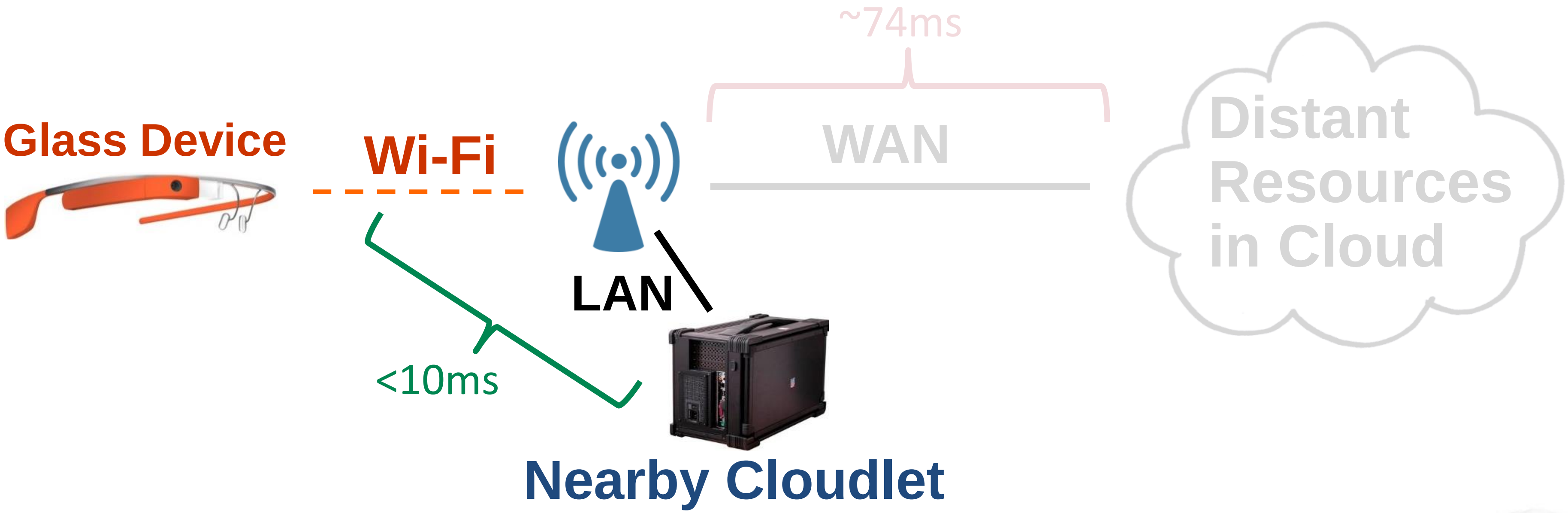
Satellite Phone?

ANNOYING!

Goal: Latency of infrastructure = tens of millisecond

S1. Crisp Interactive Response

- ✗ Choice 1: standalone apps
- ✗ Choice 2: offload to cloud
- ✓ Choice 3: offload to cloudlets



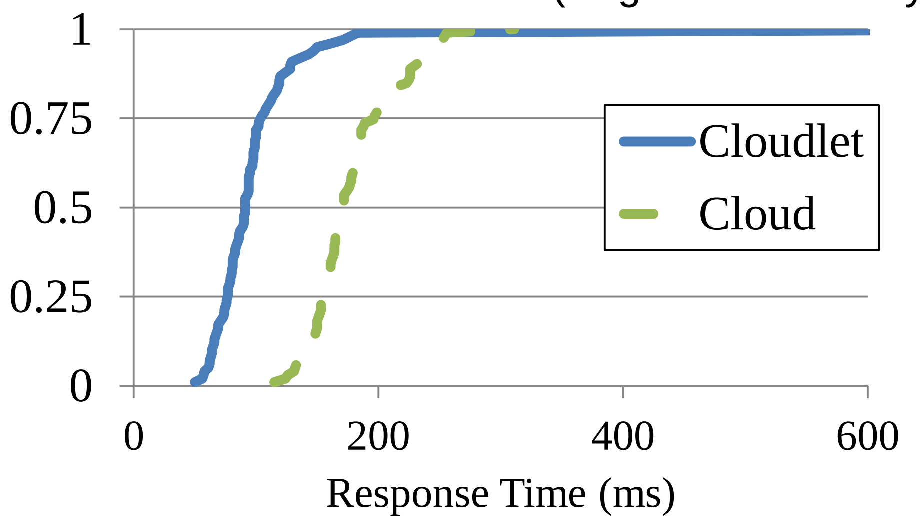
Exp. – Cloudlet Shortens Latency

Offloading saves latency and energy

Offloading vs. Standalone (OCR)

Metric	Standalone	With Offload
Per-image speed (s)	10.49	1.28
Per-image energy(J)	12.84	1.14

Cloudlet vs. Cloud (Augmented Reality)



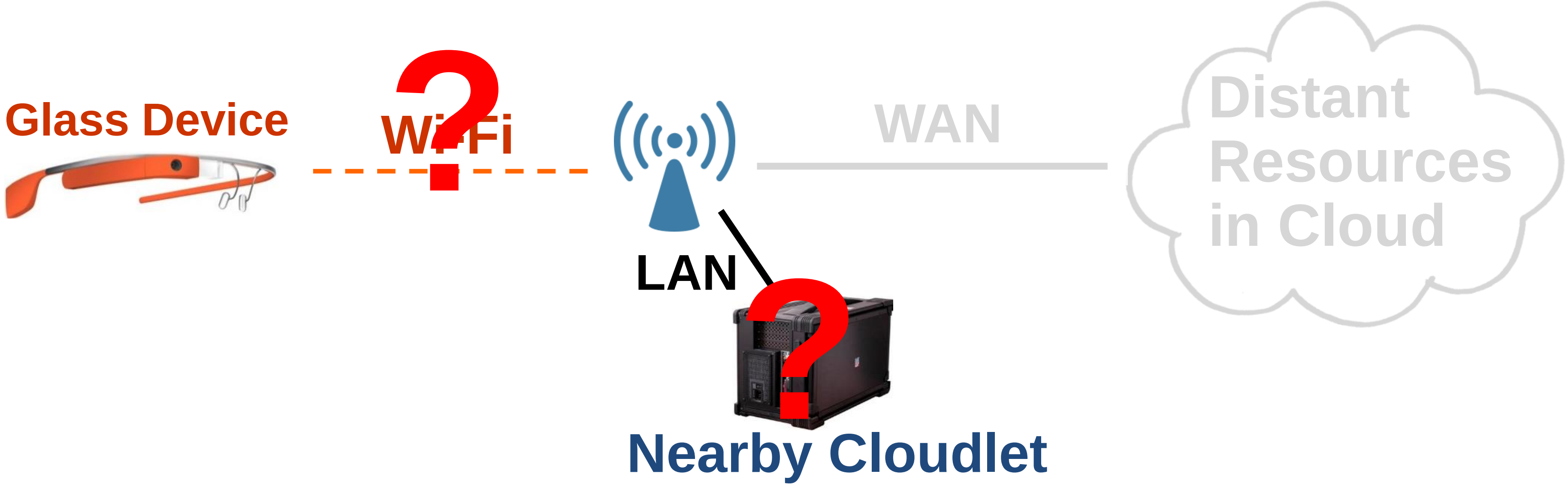
Cloudlet shortens response time

C2. Graceful Degradation of Services

What if offloading impossible?

Situation 1: No cloudlet

Situation 2: No network



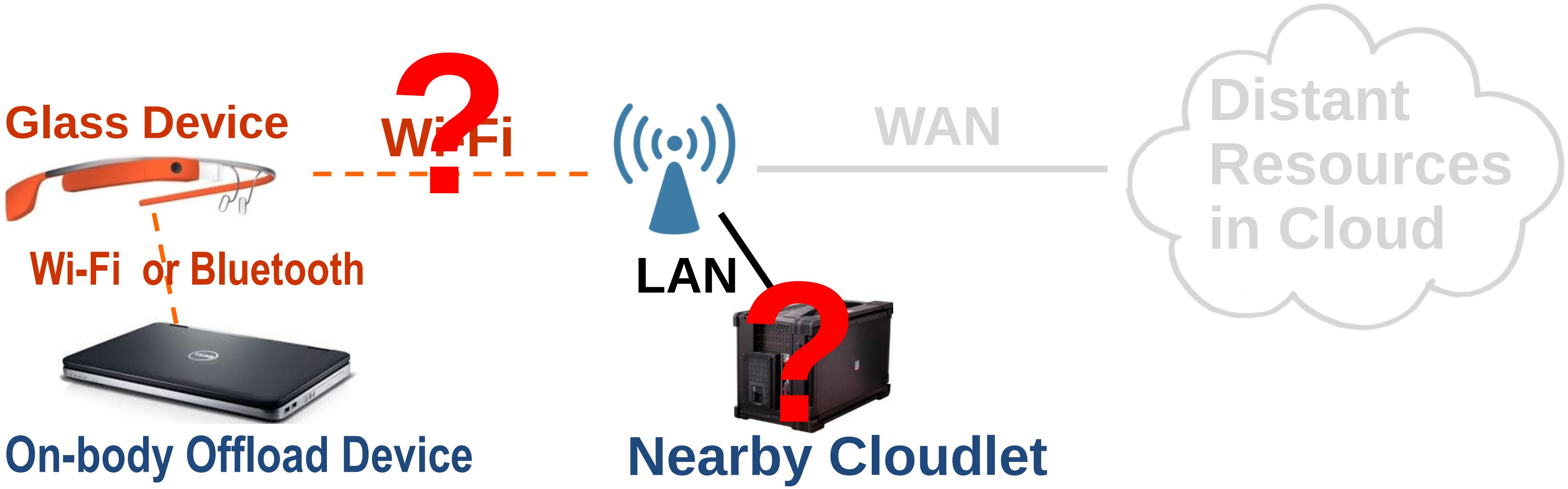
Goal: still work during failures – with performance drop

S2. Graceful Degradation of Services

Use fallback resources

No cloudlet

No network



Glass Device

Wi-Fi

?

WAN

Distant Resources in Cloud

Wi-Fi or Bluetooth

LAN

?

On-body Offload Device

Nearby Cloudlet

Application-specific fidelity vs. Crispness & battery life

C3. Coarse-grain Parallelism

Don’t reinvent the wheel

Face recognition
Object detection
Activity inference
OCR ……

Goal: reuse existing work, but…

- Programming languages are different

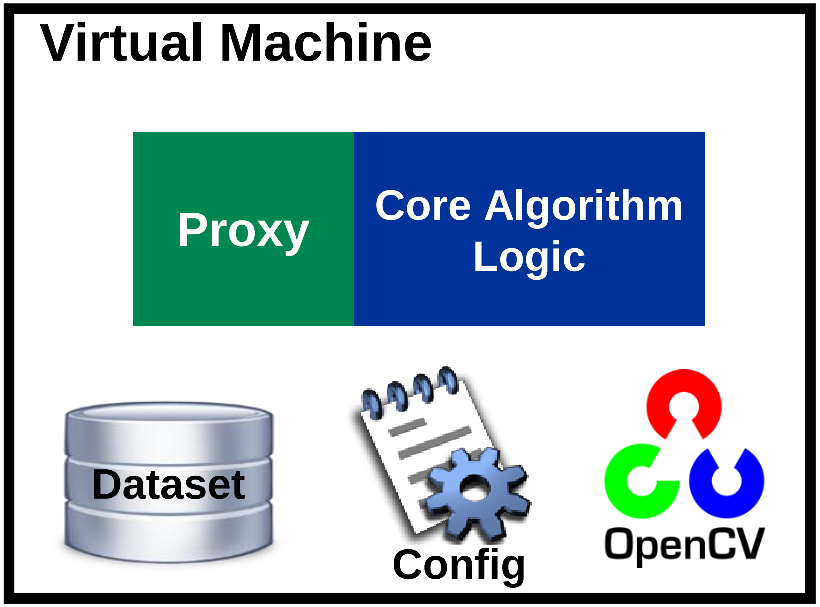
C Language Java python MATLAB

- Runtime systems are different (different OSes, closed-source, etc.)

Microsoft Windows Linux Mac OS

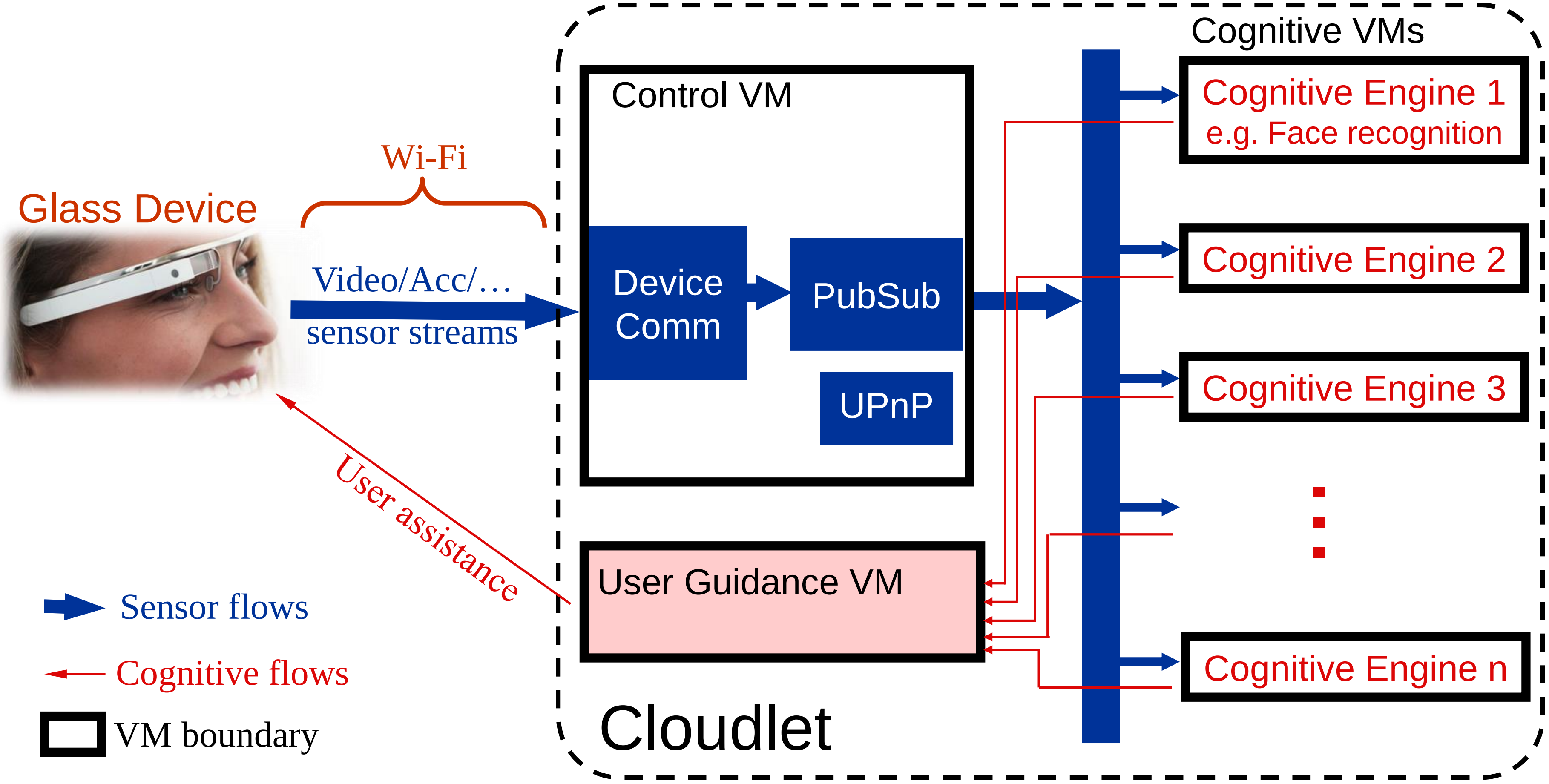
S3. Coarse-grain Parallelism

VM Ensemble and PubSub Backbone



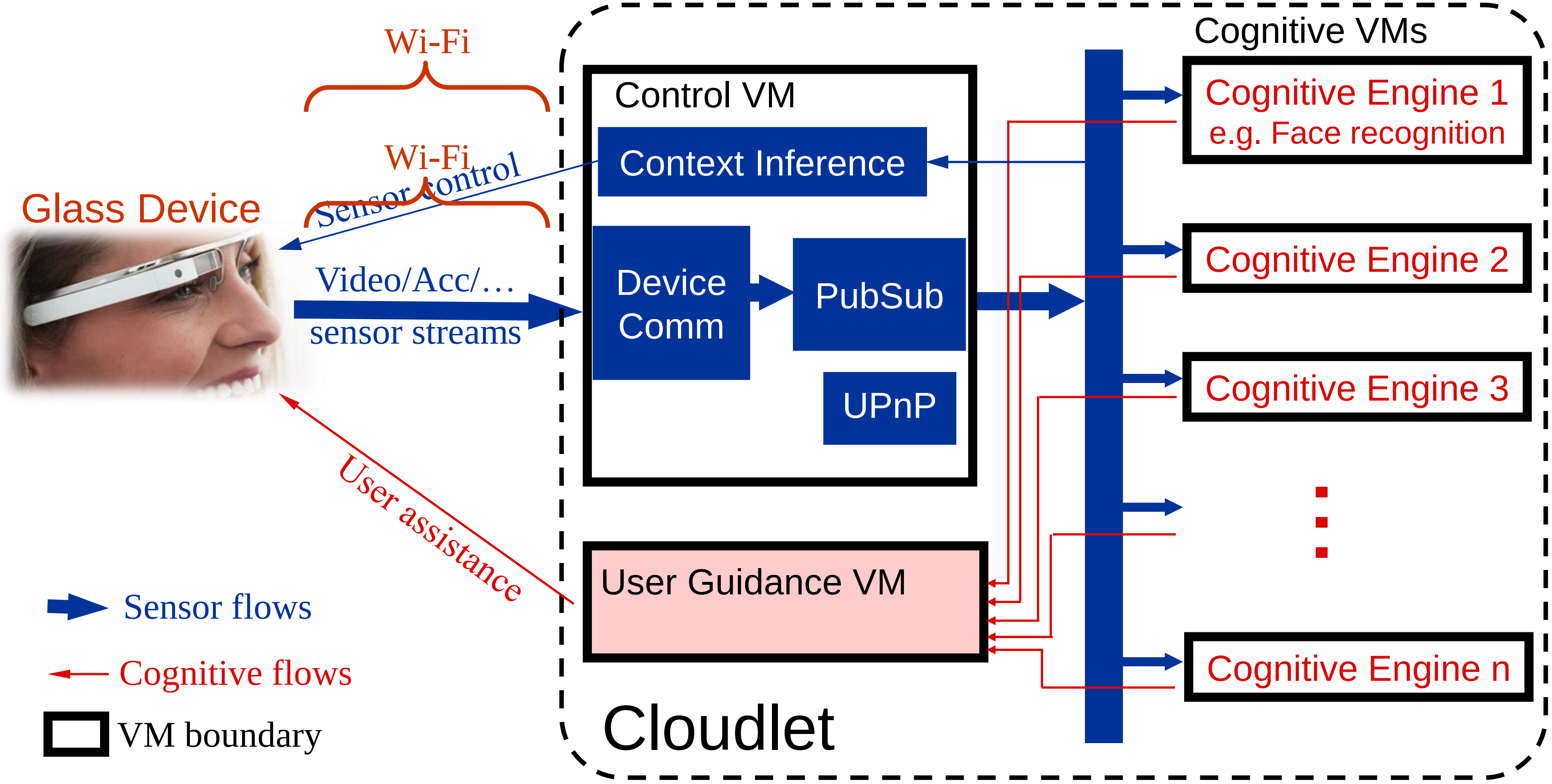
S3. Coarse-grain Parallelism

VM Ensemble and PubSub Backbone



S3. Coarse-grain Parallelism

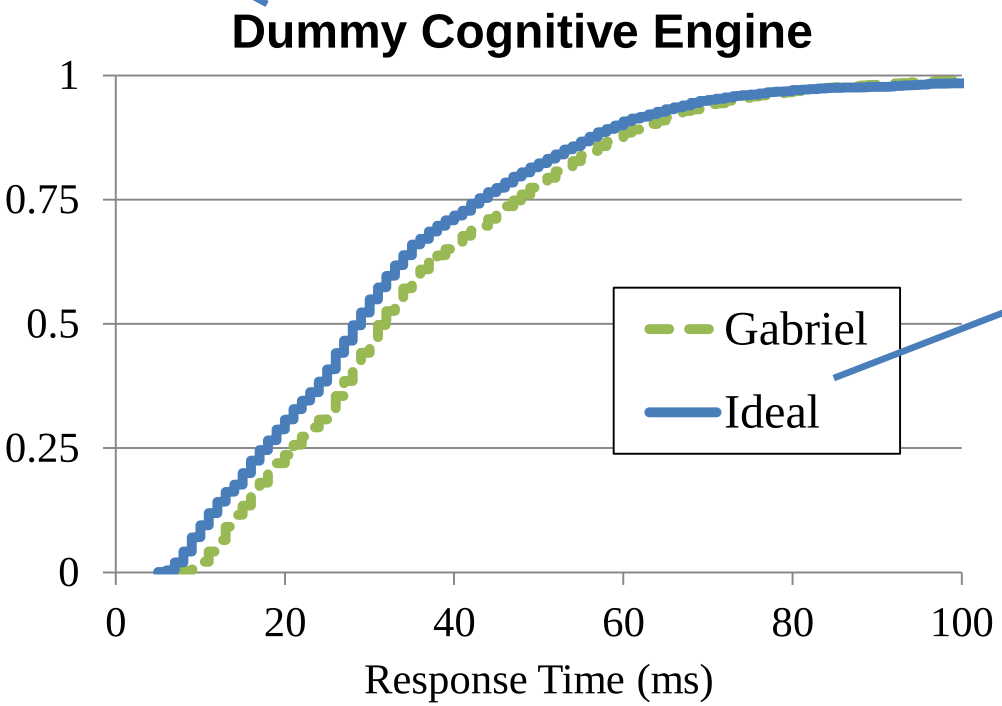
VM Ensemble and PubSub Backbone



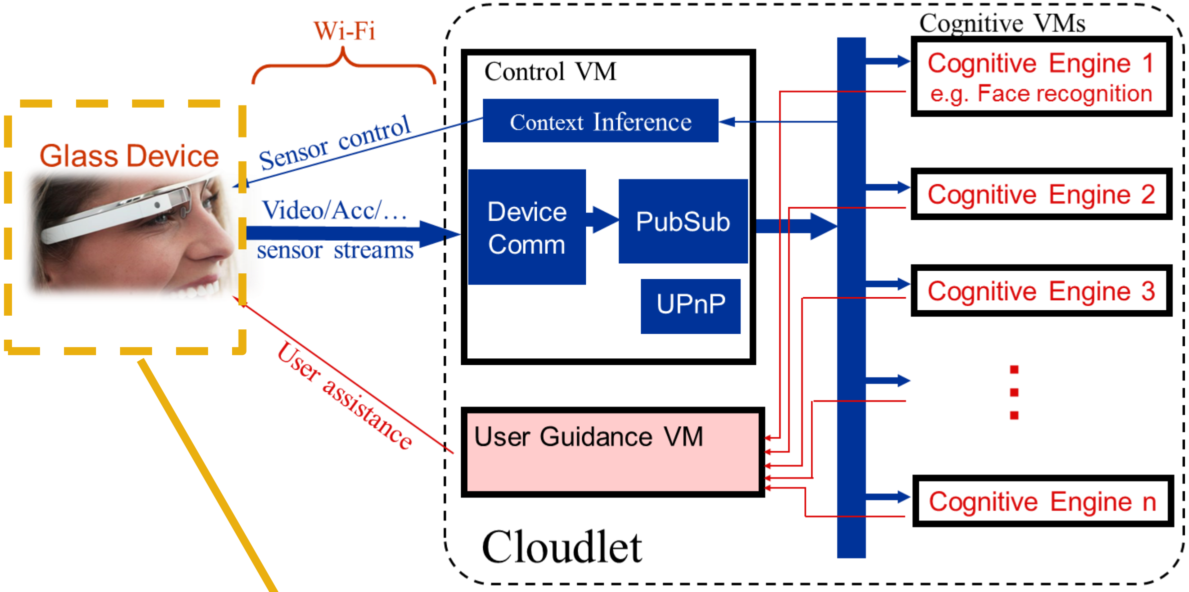
Exp. – Gabriel Overhead

Gabriel Latency: ~50ms
Gabriel Overhead: ~3ms

Echos every image



No VM, echos from host



Prototype

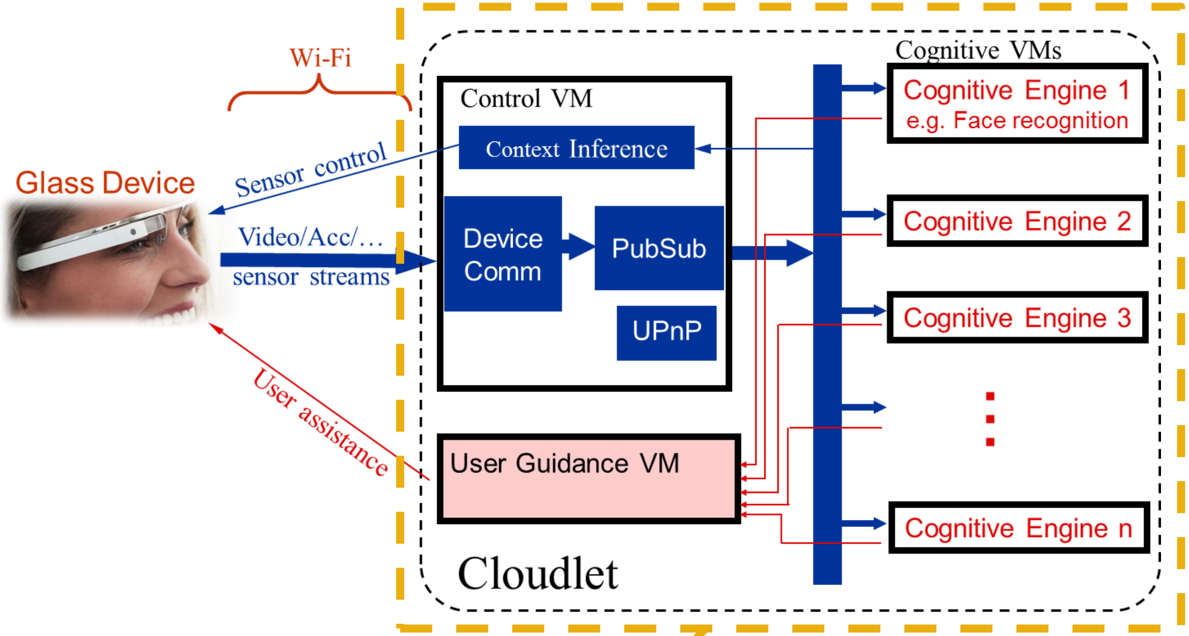
Back-end Server

GDK Preview

TCP Connection

Speech Guidance

Ice pack to cool down Glass

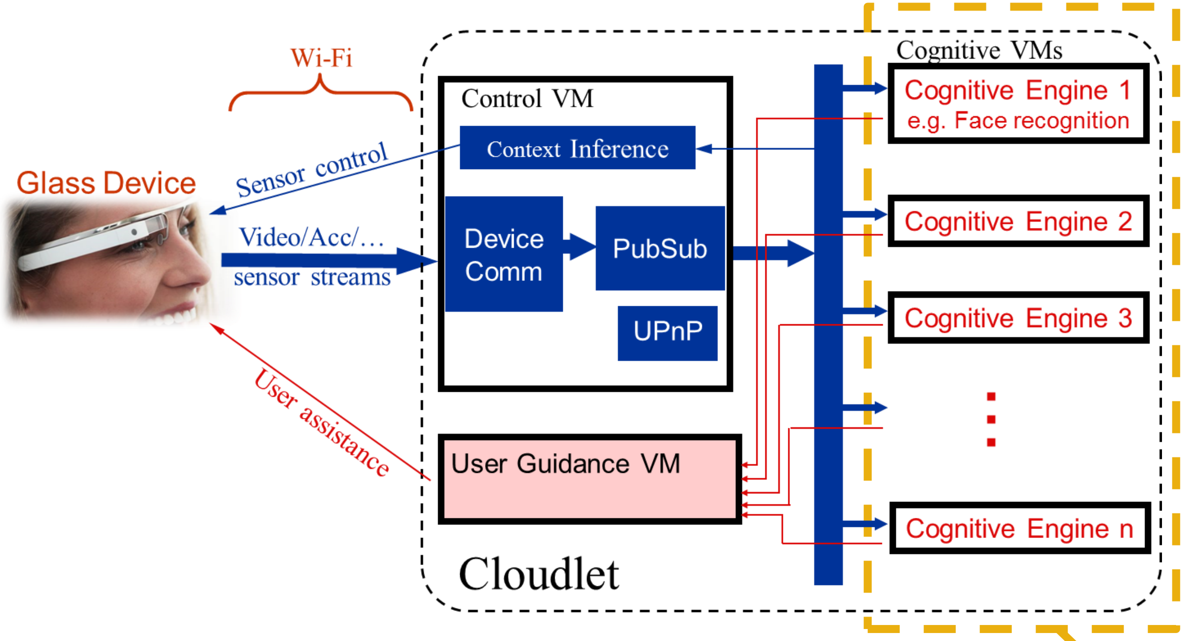


Prototype

Back-end Server

Cloudlet: 4 advanced desktop machines

Running OpenStack – Virtualized Cloud Computing Platform



Prototype

Cognitive Engines

Microsoft Windows

Linux

C Language

C++

python

Face Recognition
Object Recognition (1. MOPED 2. STF)
OCR (1. Tesseract 2 VeryPDF)
Motion Classifier
Augmented Reality
Activity Detection

Commercial Product

Based on Accelerometer

Exp. – Full System Performance

Cognitive Engines are slower

Cognitive Engine	FPS	Response time (ms)					Glass Life
		1%	10%	50%	90%	99%	
Face Recognition	4.4	196	389	659	929	1175	~1 hour
Object (MOPED)	1.6	877	962	1207	1647	2118	
Object (STF)	0.4	4202	4371	4609	5055	5684	
OCR (Open)	14.4	29	41	87	147	511	
OCR (Comm)	2.3	394	435	522	653	1021	
Motion Classifier	14.0	126	152	199	260	649	
Augmented Reality	14.1	48	72	126	192	498	

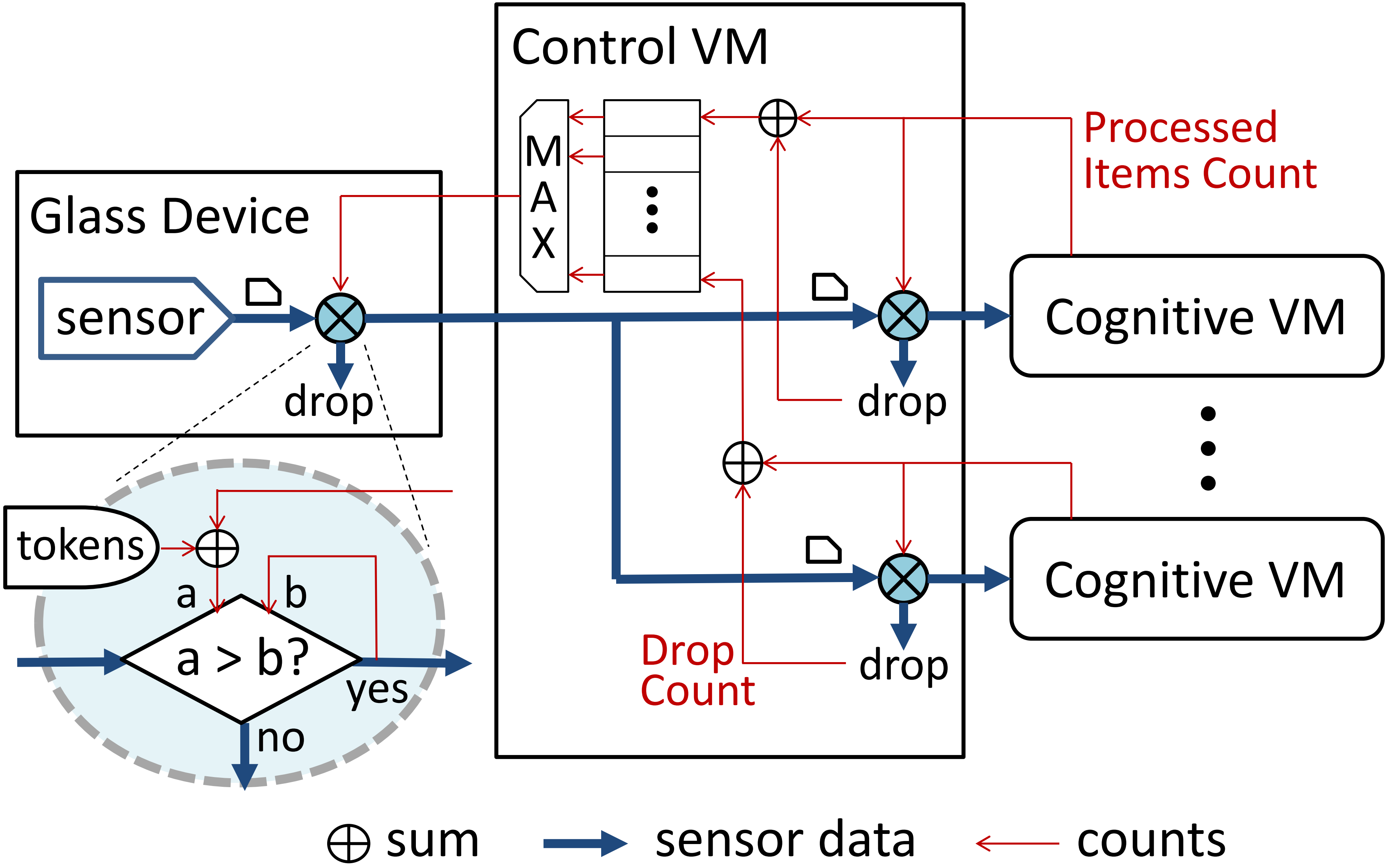
Gabriel architecture allows easy upgrade.

Exp. – Full System Performance

Cognitive Engines require different FPS

Cognitive Engine	FPS	Response time (ms)					Glass Life
		1%	10%	50%	90%	99%	
Face Recognition	4.4	196	389	659	929	1175	~1 hour
Object (MOPED)	1.6	877	962	1207	1647	2118	
Object (STF)	0.4	4202	4371	4609	5055	5684	
OCR (Open)	14.4	29	41	87	147	511	
OCR (Comm)	2.3	394	435	522	653	1021	
Motion Classifier	14.0	126	152	199	260	649	
Augmented Reality	14.1	48	72	126	192	498	

Gabriel uses **two-level token-based flow control**


Control VM
MAX
Processed Items Count
Glass Device
sensor
drop
Cognitive VM
drop
Cognitive VM
tokens
a
b
a > b?
yes
no
Drop Count
drop
sum
sensor data
counts

More in the Paper

1. Token-based flow control improves response time a lot

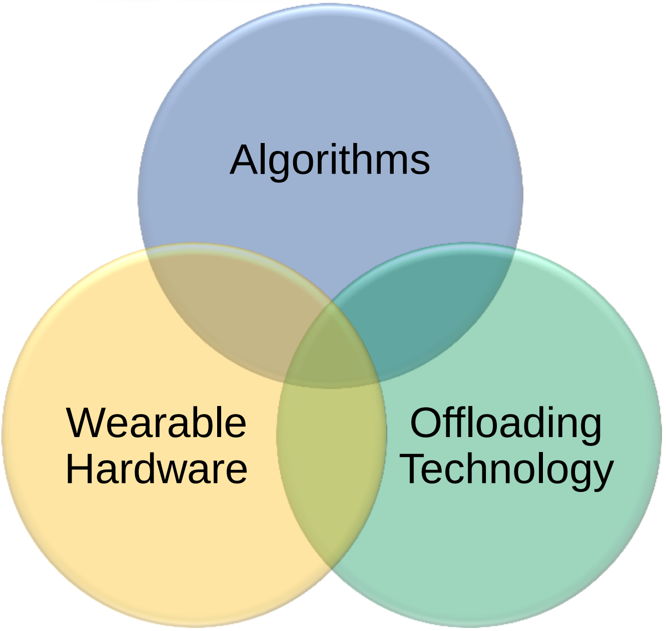
2. Gabriel supports multi-VM parallelism

3. Tradeoff between fidelity reduction and crisp user interaction

Conclusion & Future Work

Gabriel: low-latency, flexible architecture

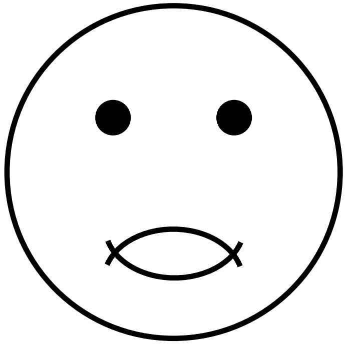
Speed improvement needed



Longer battery, better thermal dissipation

Cloudlets are helpful, need good biz. model

Gabriel:
Towards Wearable Cognitive Assistance



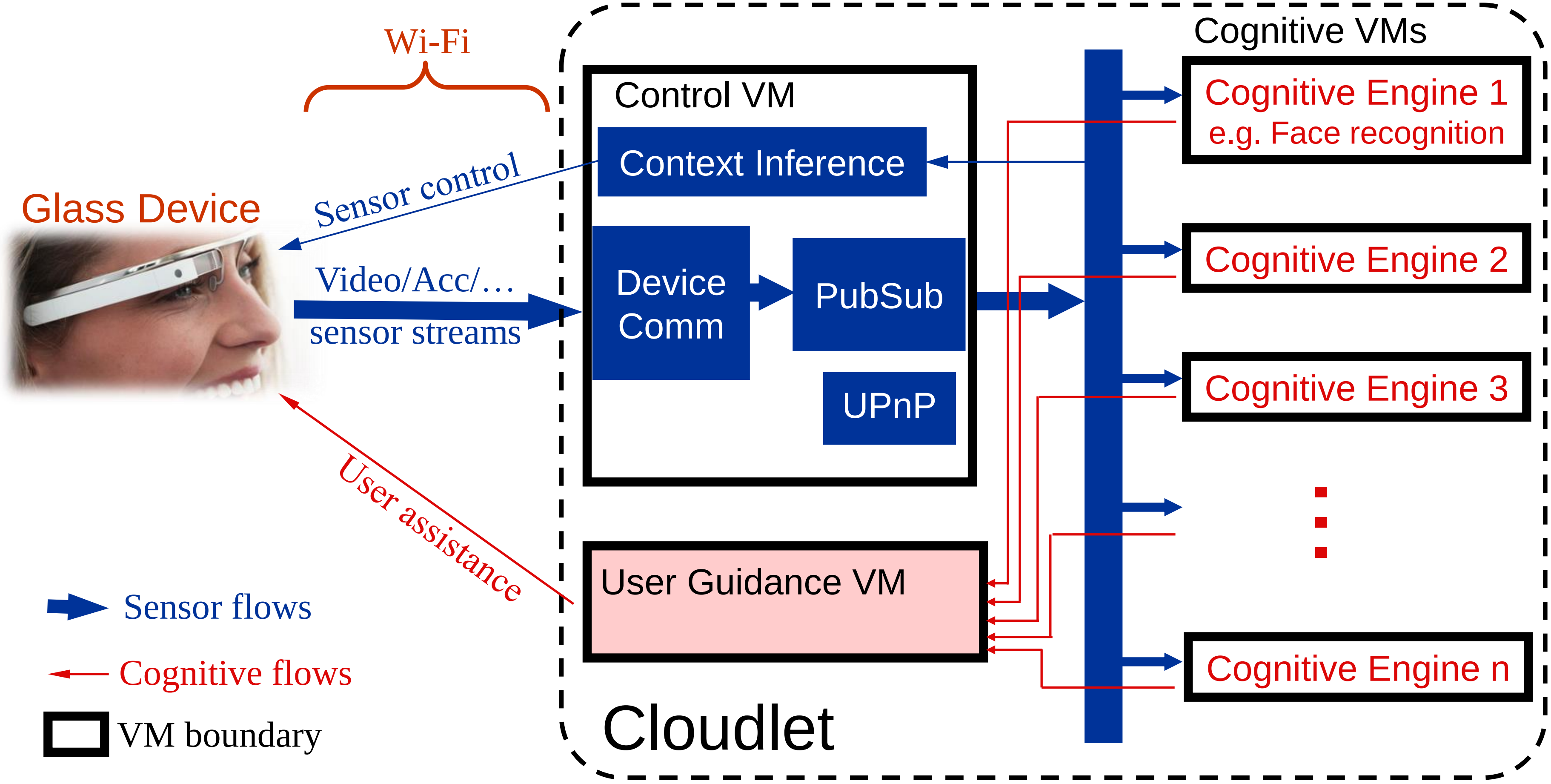
Contact: Kiryong Ha (krha@cmu.edu) Zhuo Chen(zhuoc@cs.cmu.edu)

Carnegie Mellon University intel

Backup

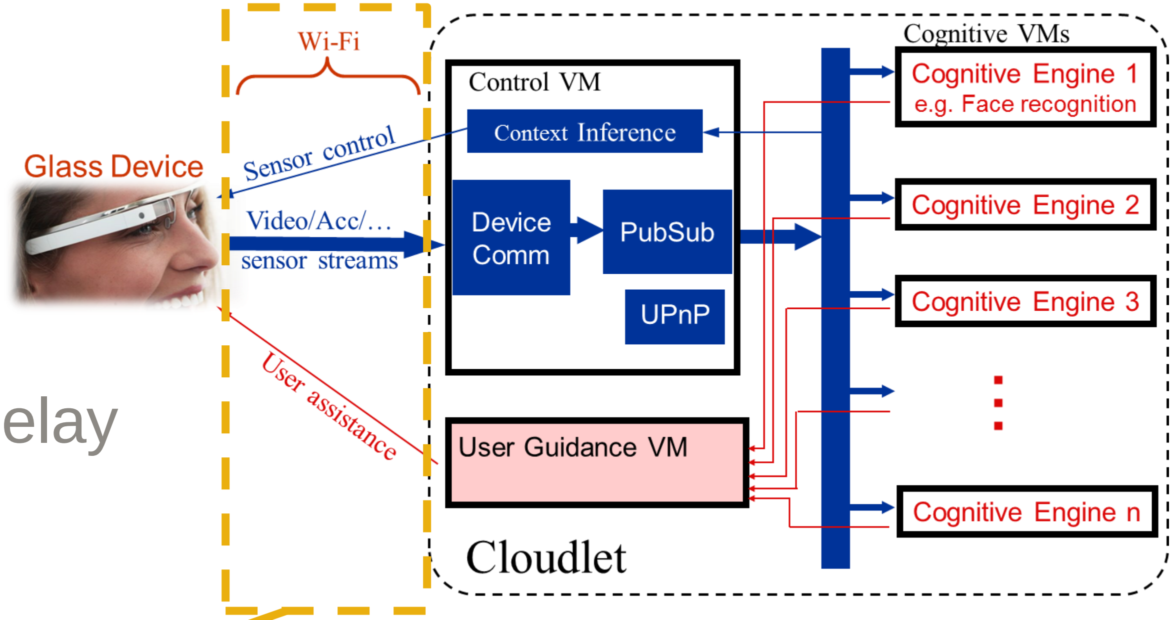
S3. Coarse-grain Parallelism

VM Ensemble and PubSub Backbone



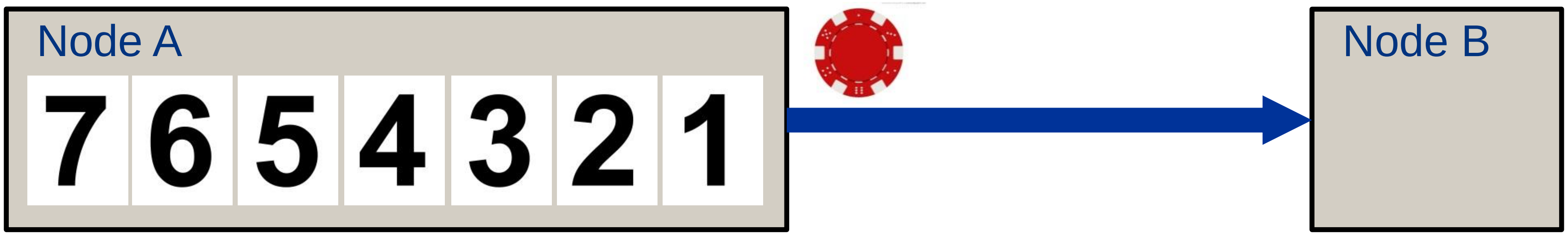
Prototype

Mitigating Queueing Delay



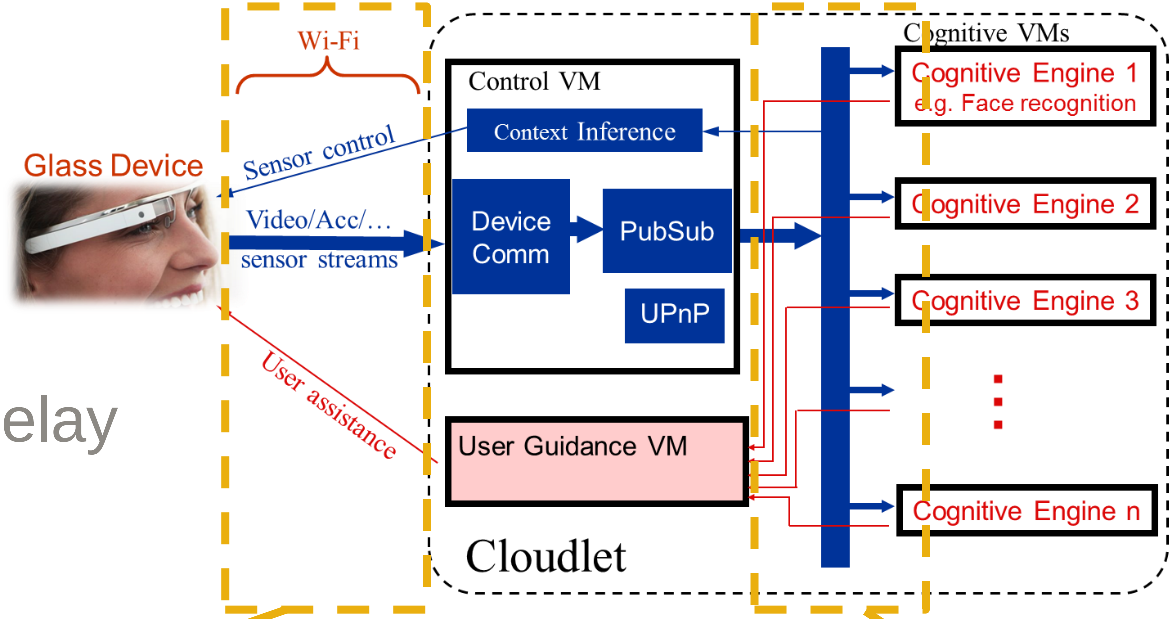
Queues hurt latency:

token-based flow control



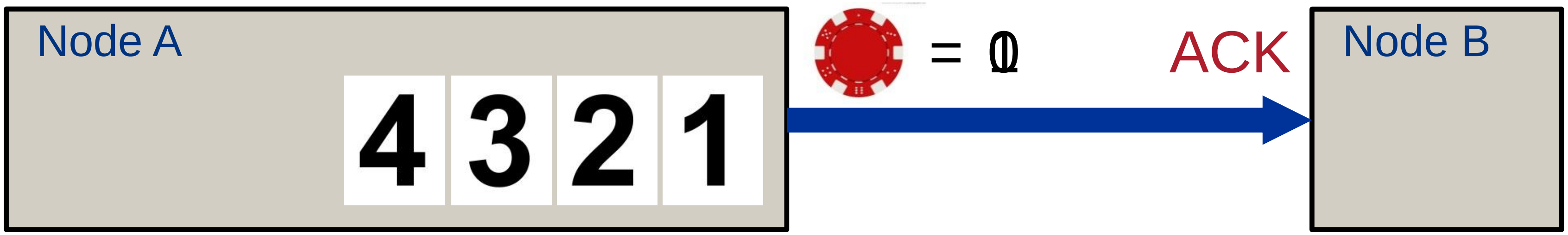
Prototype

Mitigating Queueing Delay



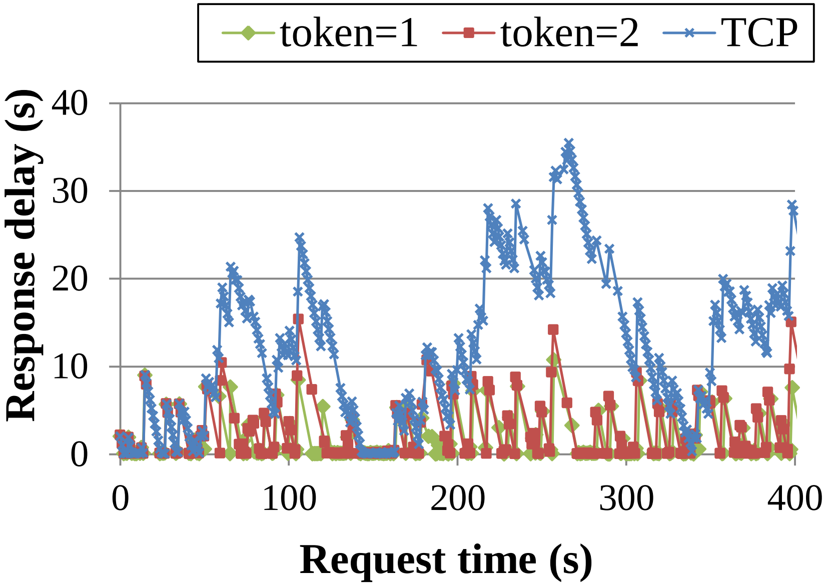
Queues hurt latency:

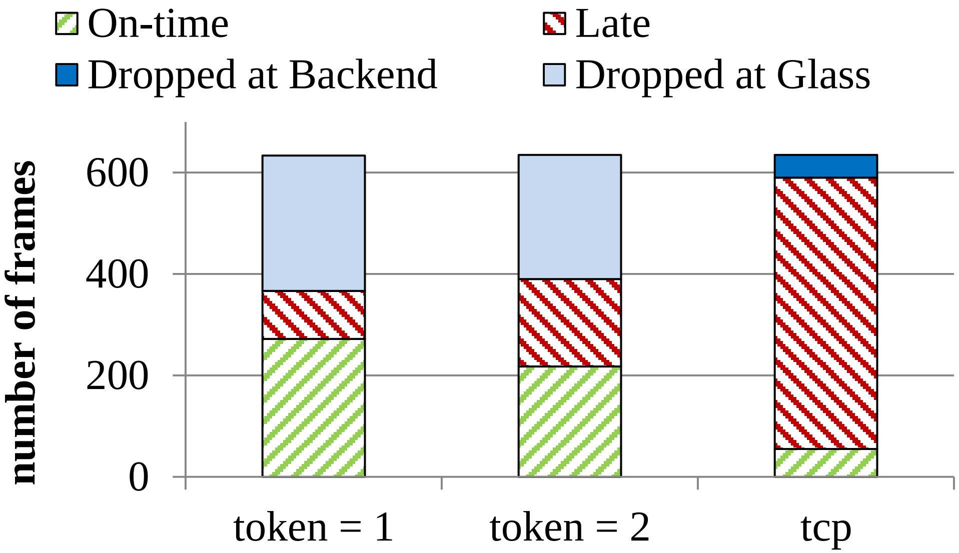
token-based flow control



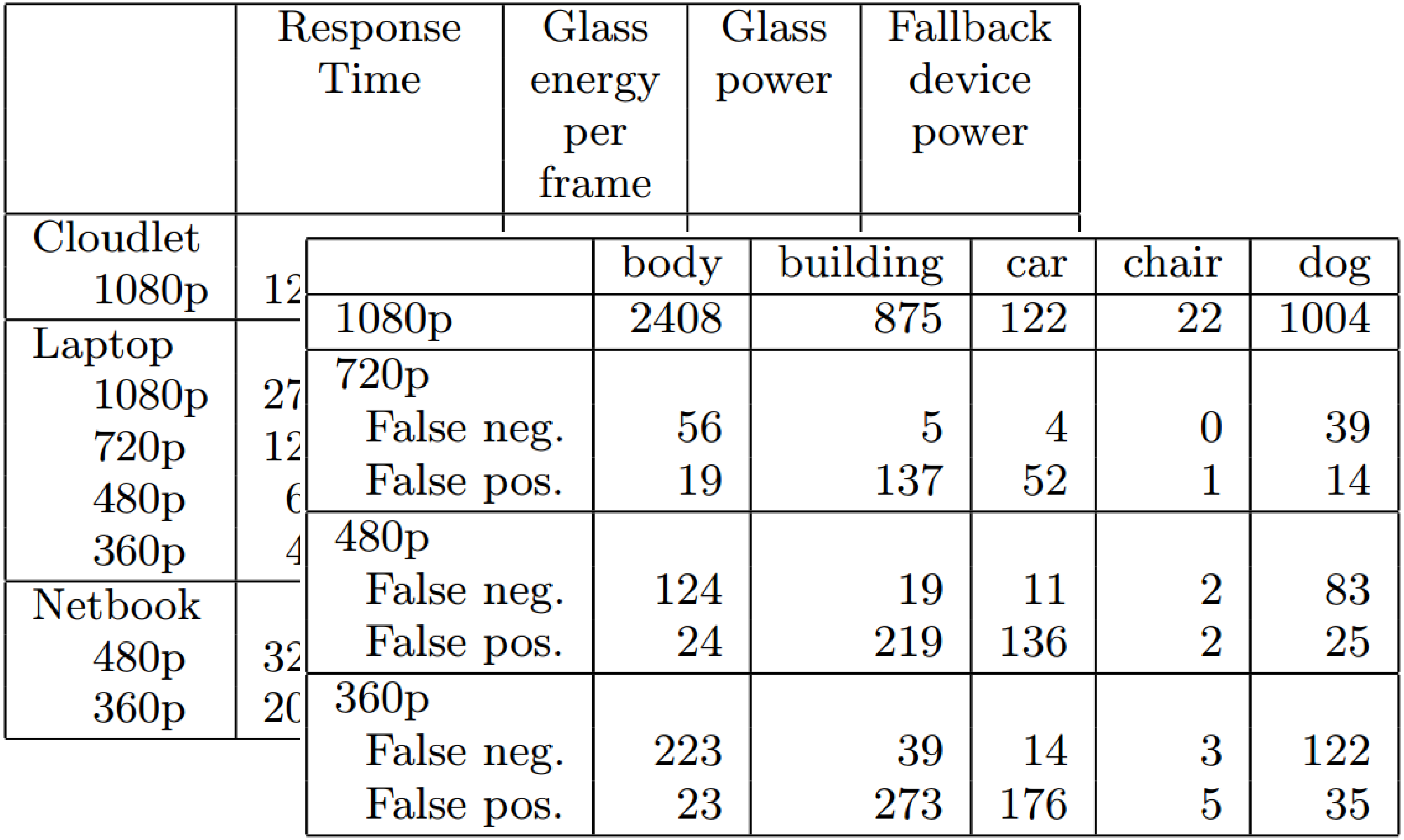
Throughput vs. Latency

Exp. – Queuing Delay Mitigation





Fidelity vs. Response Time



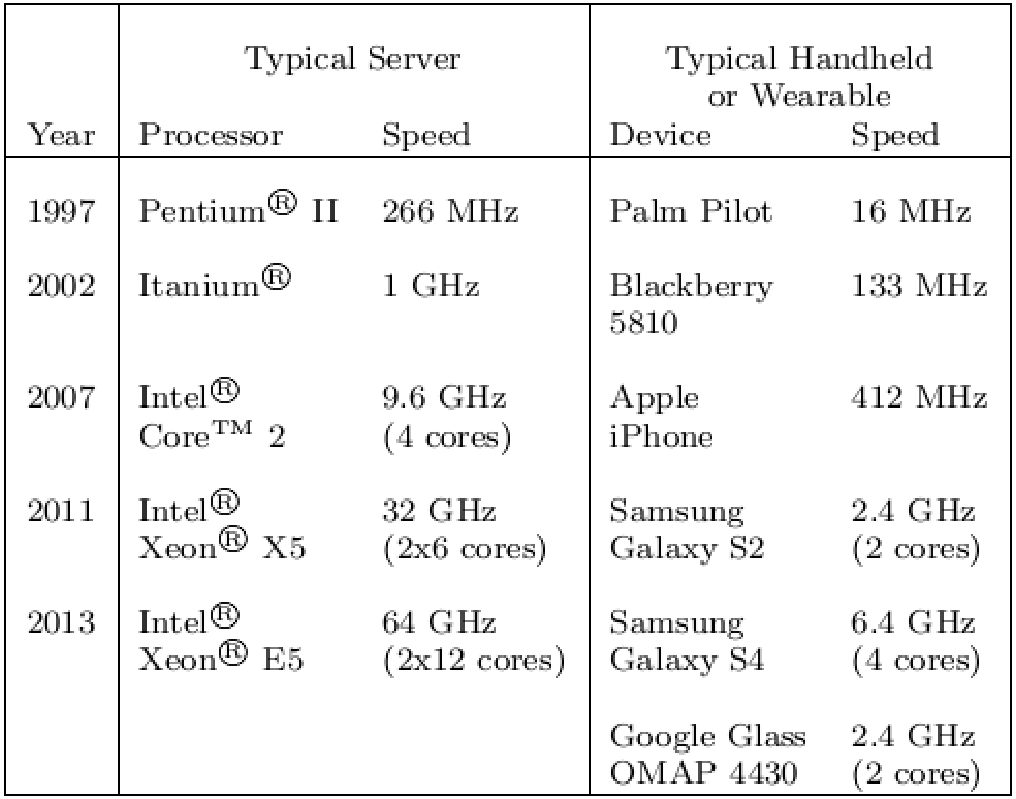
	Response Time	Glass energy per frame	Glass power	Fallback device power
Cloudlet				
1080p	12			
Laptop				
1080p	27			
720p	12			
480p	6			
360p	4			
Netbook				
480p	32			
360p	20			

	body	building	car	chair	dog
1080p	2408	875	122	22	1004
720p					
False neg.	56	5	4	0	39
False pos.	19	137	52	1	14
480p					
False neg.	124	19	11	2	83
False pos.	24	219	136	2	25
360p					
False neg.	223	39	14	3	122
False pos.	23	273	176	5	35

S1. Crisp Interactive Response

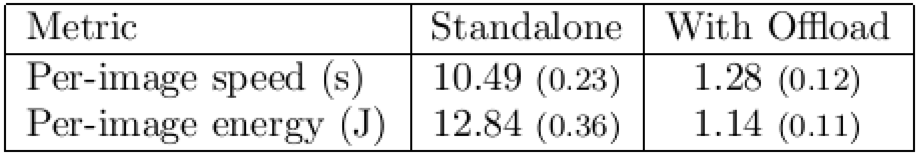
Justification of offloading

Table 1: Evolution of hardware performance



Year	Typical Server		Typical Handheld or Wearable	
	Processor	Speed	Device	Speed
1997	Pentium® II	266 MHz	Palm Pilot	16 MHz
2002	Itanium®	1 GHz	Blackberry 5810	133 MHz
2007	Intel® Core™ 2	9.6 GHz (4 cores)	Apple iPhone	412 MHz
2011	Intel® Xeon® X5	32 GHz (2x6 cores)	Samsung Galaxy S2	2.4 GHz (2 cores)
2013	Intel® Xeon® E5	64 GHz (2x12 cores)	Samsung Galaxy S4	6.4 GHz (4 cores)
			Google Glass OMAP 4430	2.4 GHz (2 cores)

Table 2: Experiment result with OCR



Metric	Standalone	With Offload
Per-image speed (s)	10.49 (0.23)	1.28 (0.12)
Per-image energy (J)	12.84 (0.36)	1.14 (0.11)

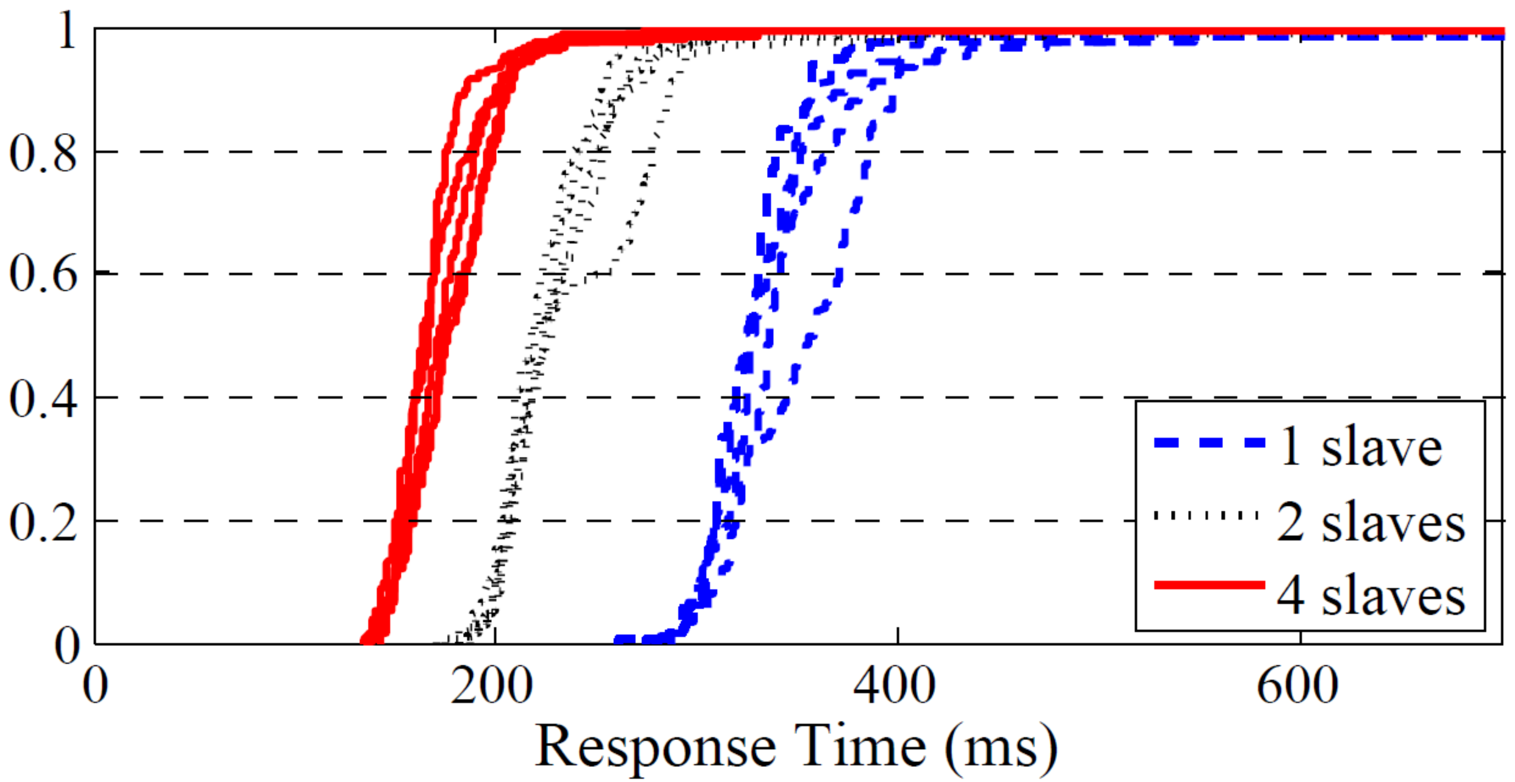
Scale-Out of Cognitive Engines

- Cognitive Engine: Motion Classifier
  - Slaves run in separate VMs

- Feature extraction

- Classification
  - A subset of classes for each slave
    - Waving
    - Clapping
    - …

Scale-Out of Cognitive Engines



# of slaves	1	2	4
Frame Rate / fps	9.8	15.9	19.0