Aalto University
School of Science

Mobile Sensing: From Personal Sensing to Crowdsensing

Yu Xiao
Department of Computer Science and Engineering
Aalto University
yu.xiao@aalto.fi

Learning Objectives

- Get to know
  - What kind of information you can get from different sensors in mobile consumer devices
  - Typical sensing applications
  - Alternative deployment models of mobile sensing applications

- To understand
  - The difference between personal sensing and crowdsensing
  - The key to the success of crowdsensing applications
  - The challenges to the emerging crowdsensing applications

How many sensors can you find in your smartphone?

Sensors in Smartphones

Samsung Galaxy S4

- Dual Cameras: photo/video (1080p@30fps)
- Microphone
- Position: GPS, Wi-Fi, cellular, Bluetooth, NFC
- Accelerometer, Gyroscope, Proximity,Compass
- Barometer
- Temperature
- Humidity
- Gesture

Apple iPhone 5

- Dual Cameras: photo/video (1080p@30fps), panorama
- Microphone
- Position: GPS, Wi-Fi, Cellular, Bluetooth
- Accelerometer
- Gyroscope
- Proximity
- Compass
- Ambient light sensor

Sensors in Emerging Wearable Devices

Google Glasses

- 5-megapixel camera, 720p video recording
- Microphone
- GPS
- Wi-Fi 802.11b/g
- Bluetooth
- Gyroscope
- Accelerometer
- Compass
- Ambient light sensing and proximity sensor

Samsung Galaxy Gear

- 1.9-megapixel camera, 720p video recording
- Microphone
- Bluetooth
- Gyroscope
- Accelerometer
- Pedometer

Are these sensors programmable?

How can I collect the sensor data?

Example of getting sensor information using Android 4.2.2 SDK

```
SensorManager mSensorManager =
(SensorManager)getSystemService(SENSOR_SERVICE);
List<Sensor> mSensorList = mSensorManager.getSensorList(Sensor.TYPE_ALL);
String sResult = "";

for (Sensor mSensor: mSensorList){
    sResult += String.format("Name:%s, maxRange:%f, Resolution:%f, Power:%f;\r\n",
mSensor.getName(), mSensor.getMaximumRange(), mSensor.getResolution(),
mSensor.getPower());
}
```

```
Name:Light sensor, maxRange:10000.000000, Resolution:1.000000, Power:0.750000;
Name:Proximity sensor, maxRange:5.000000, Resolution:5.000000, Power:0.750000;
Name:L3G4200D Gyroscope sensor, maxRange:34.906586, Resolution:0.001222,
Power:6.100000;
Name:iNemo Linear Acceleration sensor, maxRange:39.226601, Resolution:0.009577,
Power:0.200000;
…
```

(sample accelerometer readings at 1Hz)

```
Sensor mAccelerometer = mAccelerometer =
mSensorManager.getDefaultSensor(Sensor.TYPE_LINEAR_ACCELERATION);

mSensorManager.registerListener(this, mAccelerometer, 1000000);
```

```
public void onSensorChanged(SensorEvent event) {

    switch (event.sensor.getType()){
    case Sensor.TYPE_ACCELEROMETER:
    case Sensor.TYPE_LINEAR_ACCELERATION:
        getAccelerometerReading(event);
    break;
    …
}
```

```
float x = event.values[0];
float y = event.values[1];
float z = event.values[2];
```

(Linear accelerometer readings)

x: 0.008565, y: -0.741598, z: 0.365707
x: -0.189842, y: -0.009332, z: -0.446790
x: 0.209022, y: -0.394663, z: 0.386457
x: -0.034229, y: 0.248693, z: -0.142323
…

What can we learn from these sensor readings?

Example

- GPS (latitude, longitude, altitude) → where you are
- Compass → Which direction you are heading to
- Gyroscope → orientation of the device
- Accelorometer → …
- Photo/Video →…
- CellID→…
- Bluetooth→…
- …

We may use the sensor data for

- Navigation
- …

BrainStorming (5 min)

Personal Sensing

- Movement patterns (e.g. walking, jogging, climbing stairs)
- Modes of transportation (e.g. cycling, driving, riding a train, taking a bus)
- Activities (e.g. shopping, dining, listening to music)

Google Activity Recognition APIs

- Detect user's current physical activity
  - Walking, Still, Cycling, In vehicle
- Requests for updates go through an activity recognition client

```
ActivityRecognitionClient mActivityRecognitionClient;
mActivityRecognitionClient = new ActivityRecognitionClient(context, this, this);
mActivityRecognitionClient.connect();
```

```
public void onConnected(Bundle arg0) {

//create the PendingIntent that location service uses to send activity recognition updates
Intent intent = new Intent(context, ActivityRecognitionService.class);

mActivityRecognitionPendingIntent = PendingIntent.getService(context, 0, intent,
PendingIntent.FLAG_UPDATE_CURRENT);

mActivityRecognitionClient.requestActivityUpdates(updateIntervalInSeconds,
mActivityRecognitionPendingIntent);
}
```

- Each update includes one to multiple possible activities and the confidence level of each activity

```
public class ActivityRecognitionService extends IntentService {
@Override
protected void onHandleIntent(Intent intent) {

        if(ActivityRecognitionResult.hasResult(intent)){
                ActivityRecognitionResult result =
                        ActivityRecognitionResult.extractResult(intent);

                List<DetectedActivity> list = result.getProbableActivities();
                …
        }
}
```

- Does not require GPS or network connection
- Closed source algorithms
- Accuracy is not high

A Case of Transport Activity Survey using Smartphones

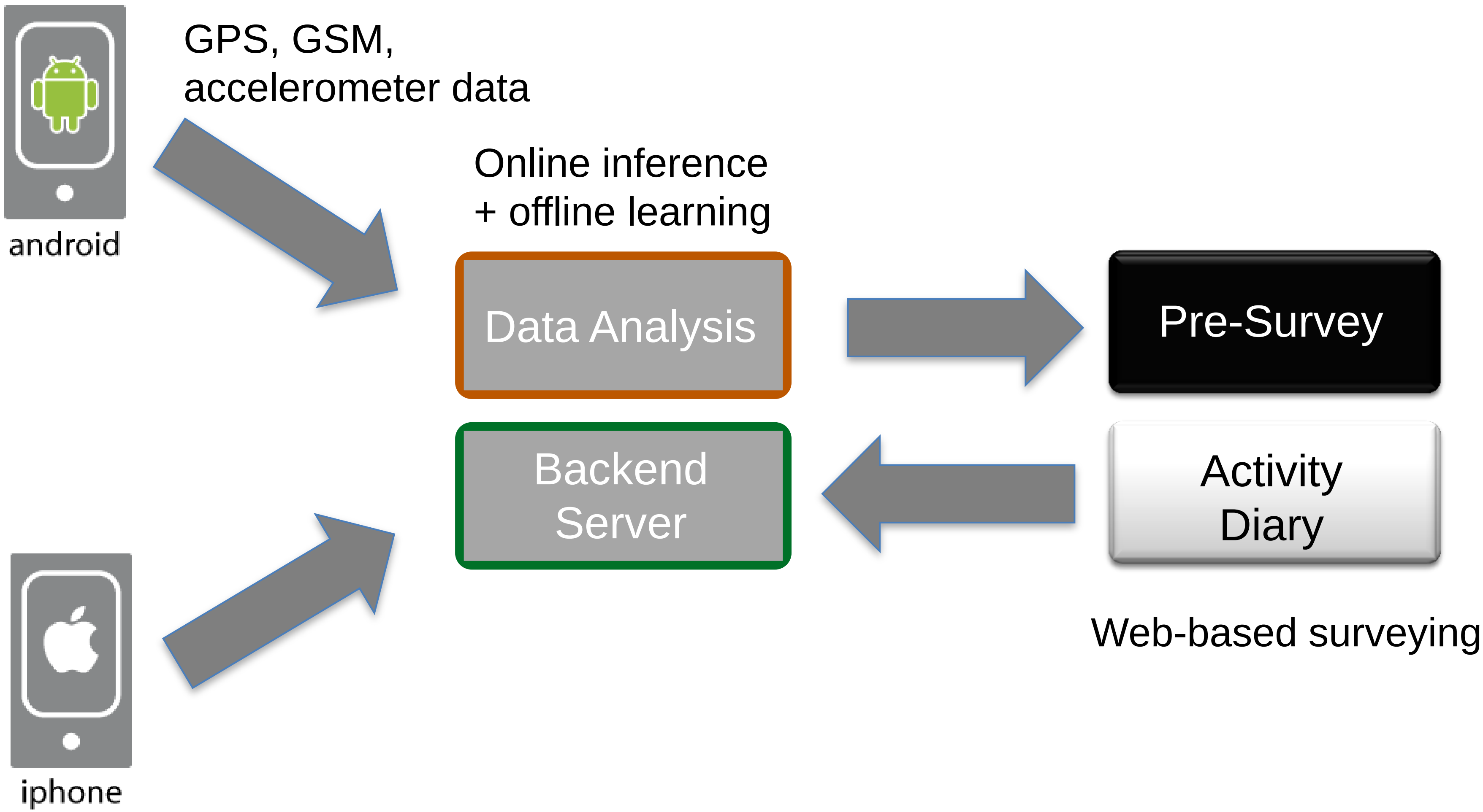
- Future Urban Mobility Programme (01.07.2010 – 30.6.2015), Singapore-MIT Alliance for Research and Technology (SMART)

Transportation Activity Survey using Smartphones

To understand

- Where people move (Trips people make in their daily lives)
- How people move (by bus, train, taxi, …)
- Stops on each trip, related to changing in transportation mode or other activities such as dining and shopping.
- Activities performed at each stop

Transportation Activity Survey



Online Inference

- Process raw data collected from smartphones
  - GPS/GSM → Trips (where and when)
  - GPS/GSM → Stops on each trip (where and when)
  - Accelerometer, GPS/GSM →Transportation mode detection (how)
  - Context-aware activity detection (why)

Accuracy of online inference can be improved when prior knowledge is incorporated. e.g.

- Bus/train routes → Trip detection
- Place information → Stop detection, transportation mode detection

Reference: Yu Xiao; Low, D.; Bandara, T.; and et al., "Transportation activity analysis using smartphones," *2012 IEEE Consumer Communications and Networking Conference (CCNC),* pp.60,61, 14-17 Jan. 2012.

Activity Diary

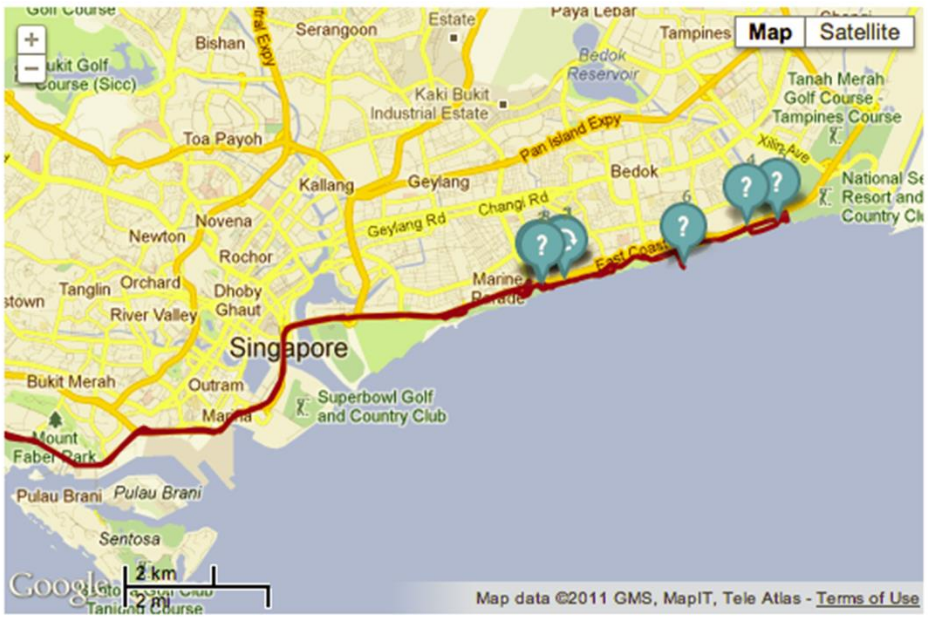
1. Member selection

2. Date Selection 20-11-2011

3. Trip Selection Trip 0 [00:03 - 16:52] - ✓

Please click here to share your opinion about selected day

4. Activity Diary



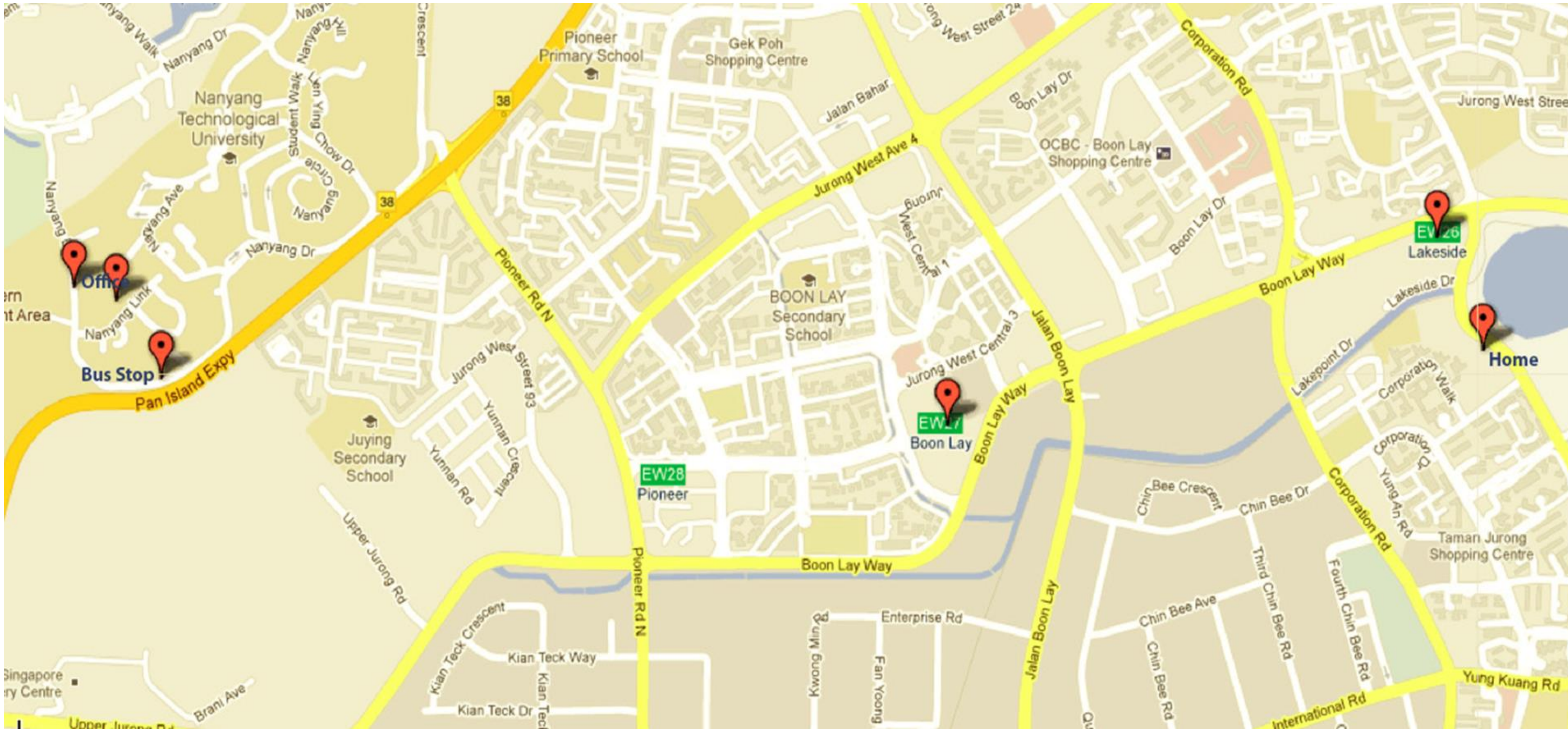
STOP NAME	START	END
Home	00:03	10:30
Bus Stop	11:00	11:02
Bus Stop	11:10	11:15
	11:22	12:11
	12:16	12:29
	12:41	12:42
Bus Stop	12:59	13:03
	13:05	16:52

Offline Learning

- Use machine learning techniques to derive contextual knowledge based on historical data and known places (such as train/bus stops)

- Contextual knowledge
  - probability distribution of activities at each stop
  - probability of different transportation modes between certain stops
  - …

My Frequently Visited Places



Density-based clustering

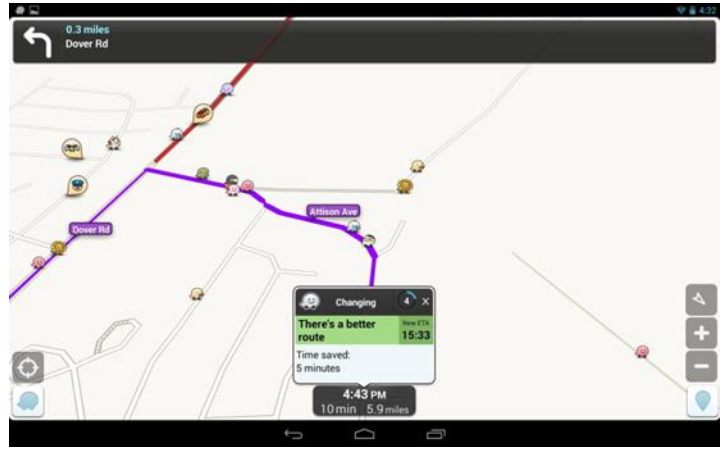
Aalto University
School of Science

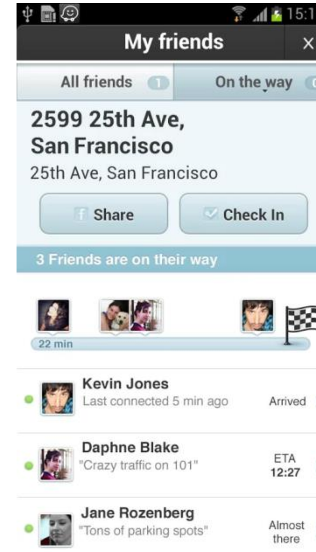
Source: http://www.cooltownstudios.com/images/crowdsourcing-cartoon.jpg

Waze Social GPS maps & traffic

http://www.youtube.com/watch?v=y_7yoEUrVhw

Transportation



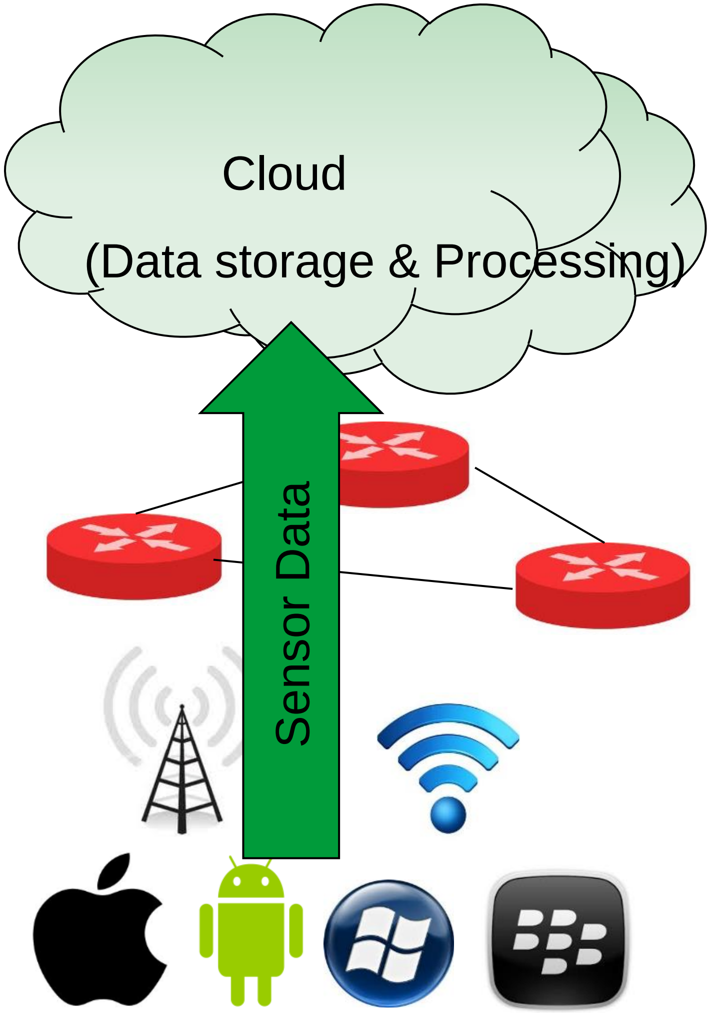


App	No. Users	Features	Sensing data
Waze social GPS maps & traffic (Waze) Android,iPhone	Over 1 million	▪ Live routing based on community-generated real-time traffic and road info. • Community-contributed road alerts including accidents, hazards, police traps, and more. • Find the cheapest gas station on your route • Share your drive on a live map, see friends also on the way to your destination.	Location (GPS and network-based) Images & Videos Your contact list

Aalto University
School of Science

Source:
https://play.google.com/store/apps/details?id=com.waze&hl=en

System Architecture



Mobile CrowdSensing Applications

Classified by the type of phenomenon being measured

- Environmental (e.g. measuring air quality and noise levels, monitoring wildlife habitats)
- Infrastructure (e.g. measuring traffic congestion, road conditions, parking availability)
- Social (e.g. share restaurant information)

Source: Ganti, R.K.; Fan Ye; Hui Lei, "Mobile crowdsensing: current state and future challenges," *Communications Magazine, IEEE* , vol.49, no.11, pp.32,39, November 2011. doi: 10.1109/MCOM.2011.6069707

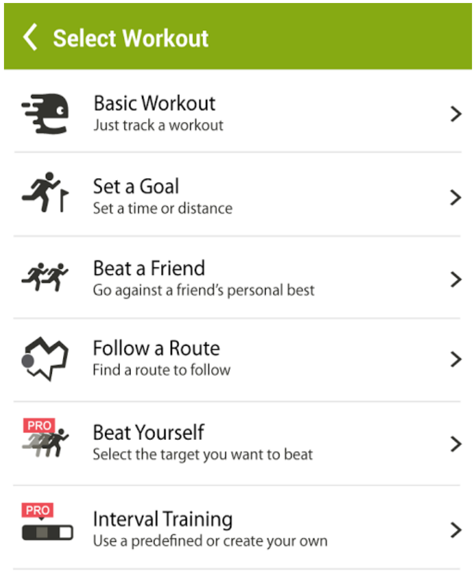
Participatory vs. Opportunistic Sensing

- Participatory Sensing
  - Individuals are actively involved in contributing sensing data (e.g. taking a photo)

- Opportunistic Sensing
  - Sensing is more autonomous and user involvement is mininal

Sports Tracker

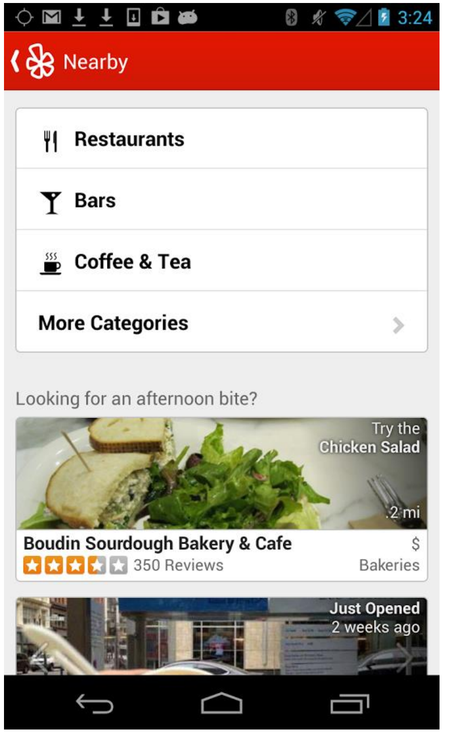
App	No. Users	Features	Sensing data
Endomondo Sports Tracker (by Endomondo) Android, Blackberry, iPhone, Windows	123163	• Track any outdoor sport including duration, distance, speed and calories • Enter a workout manually, e.g., a treadmill run or weight training. • Work with heart rate monitor sensors. • Compete on a specific route nearby and race against the route champion.	Location (GPS and network-based) Phone status & identity Your contact list



http://www.youtube.com/watch?v=3-m8Bix7OK4

Augmented Reality

App	No. Users	Features	Sensing data
Yelp (by Yelp) Android , iPhone	97927	• Search for businesses near you. • Browse reviews. • Find great Deals offered by your favorite local businesses. • Yelp does augmented reality with Monocle. Overlay business information onto the world around you.	Location (GPS and network-based) Images & Videos Your contact list



http://www.youtube.com/watch?v=e_iQra7AhRo

Aalto University
School of Science

AR: http://www.youtube.com/watch?v=D-A1I4Jn6EY

Scale is the key to the success of crowdsensing applications

Motivation & Concerns

- Motivation of joining a crowd
  - Save time
  - Save money
  - Personal healthcare
  - Public safety
  - Social networking
- Concerns (collected from user reviews)
  - Inaccurate information
  - Privacy
  - Software update
  - Battery life

Obstacles to crowdscaling

- Difficulty in finding incentive mechanism
  - Installation work
  - Sensing overhead (especially energy consumption)
  - Possible loss of privacy

Obstacles to Crowd Scaling

- Heterogeneity of sensing hardware and mobile platforms
- HTML5 is still work in progress, current situation (source: http://mobilehtml5.org/):

	Safari on iOS	Android Browser	Google Chrome	Windows 8 IE
HTML media capture (taking pictures, recording videos)	6.0+ (partially)	3.0+	Android 4.0+	Not supported
Motion Sensors (accelerometer, gyroscope)	4.2	3.0+	Chrome 30+	11+
Geolocation	supported	2.0+	supported	supported

Obstacles to Crowd Scaling

- Increasing network bandwidth demand caused by the growing usage of data-rich multimedia sensors

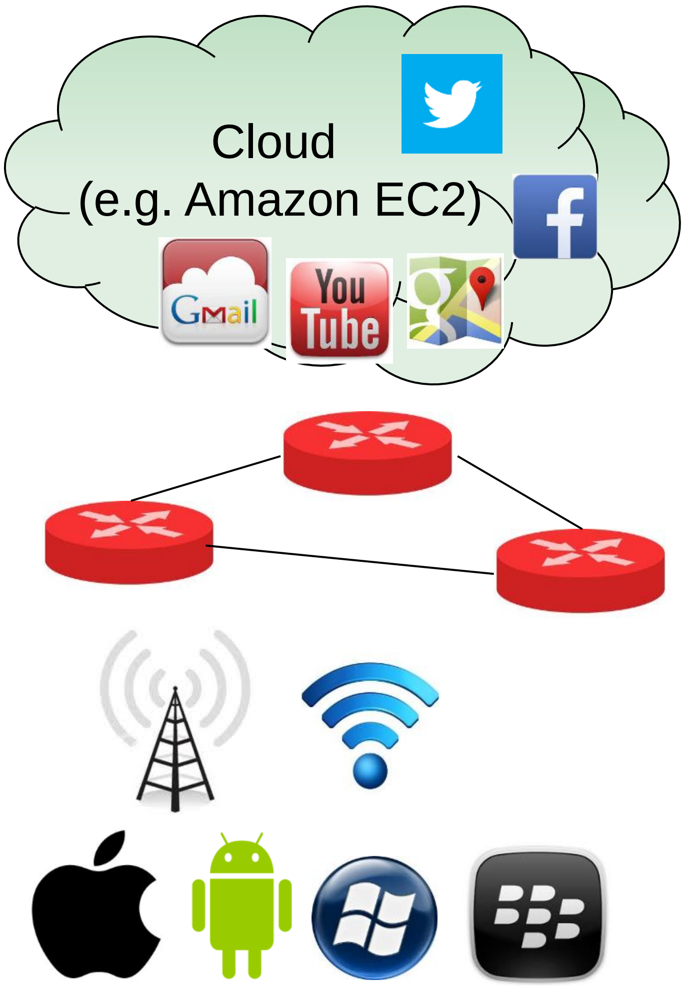
How can we lower the obstacles?

How can we lower the obstacles

- One app fits all
- Minimizing sensing overhead
- Privacy-preserving data processing

CrowdSensing in Ubiquitous Cloud Environment

Today’s Mobile Cloud Infrastucture



Centralized cloud infrastructure

- ✓ lower the marginal cost of system administration and operations

High asymmetry in traffic

- ✓ Downlink to uplink ratio: about 6:1 [1]

High latency over cellular network

- ✓ Median: 125ms[1]

Aalto University
School of Science

[1] Hossein F., Dimitrios L., Ratul M., Srikanth K., and Deborah E. 2010. A first look at traffic on smartphones. In IMC’10.

Challenges

Bandwidth-intensive

- ✓ Video to dominate the mobile data traffic in few years
- ✓ Rapid growth of user-generated content such as first-vision videos

Latency-sensitive

- ✓ 100ms is the way too long for interactive applications like augmented reality

Privacy-sensitive

- ✓ In the scenarios of crowdsensing, the value of the sensing data heavily depends on the granularity of the data
- ✓ There is a tradeoff between privacy and the value of the data

Potential Solutions

~~Bandwidth-intensive~~

Caching at the Edge vs. Increase Core Network Capacity

~~Privacy-sensitive~~

Virtualization, crowdsourcing: 1+1 > 2

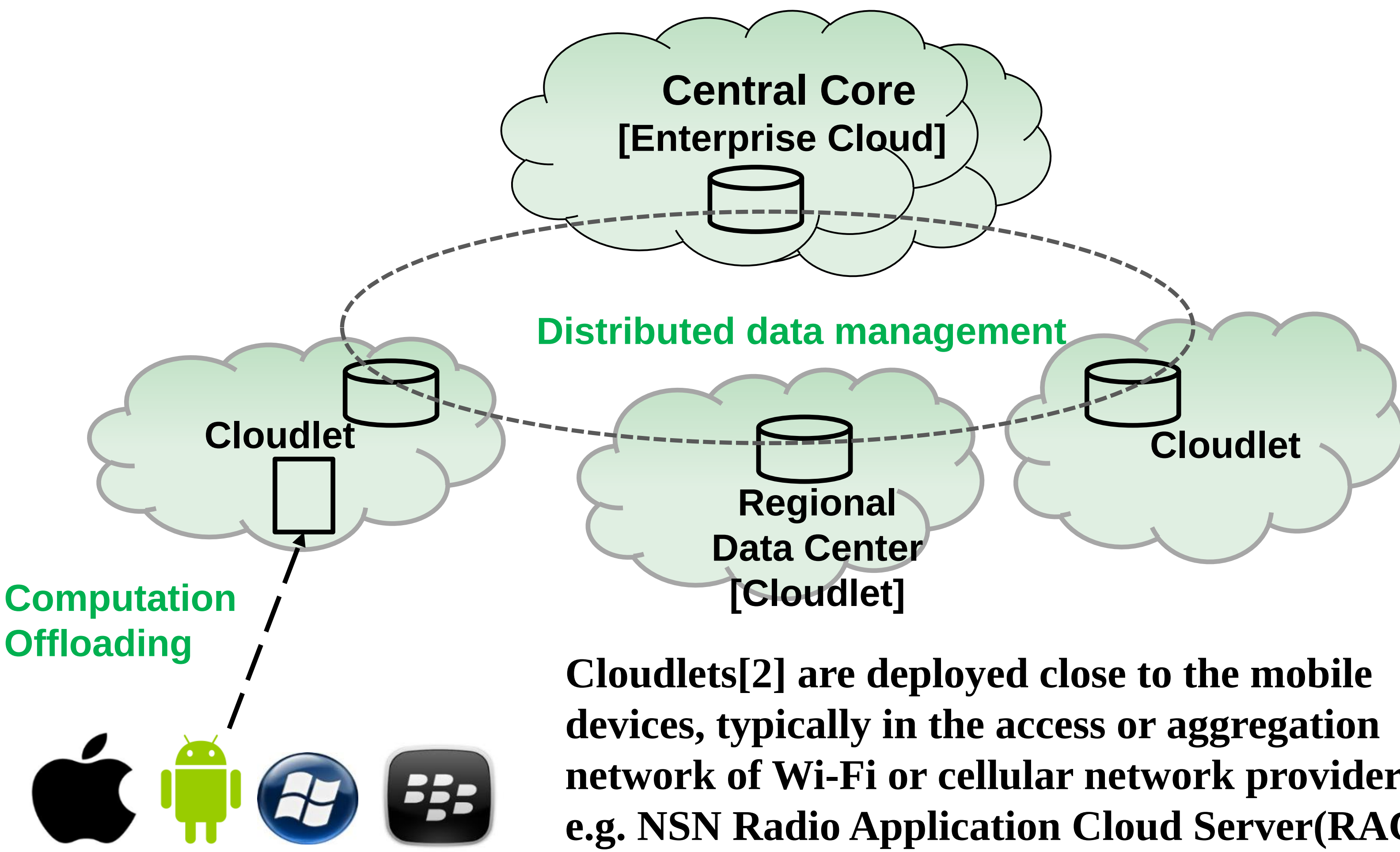
~~Latency-sensitive~~

Low latency access network + Computing at the Edge

Our Vision

The design of cloud infrastructure will move from centralization to ***ubiqutious***

Ubiquitous Cloud Infrastructure



Cloudlets[2] are deployed close to the mobile devices, typically in the access or aggregation network of Wi-Fi or cellular network providers e.g. NSN Radio Application Cloud Server(RACS) can be considered as a cloudlet collocated with LTE base station

Example Implementation of Ubiquitous Cloud Infrastructure

- The 3-tier cloud infrastructure, mobile-cloudlet-cloud, was first proposed in 2009

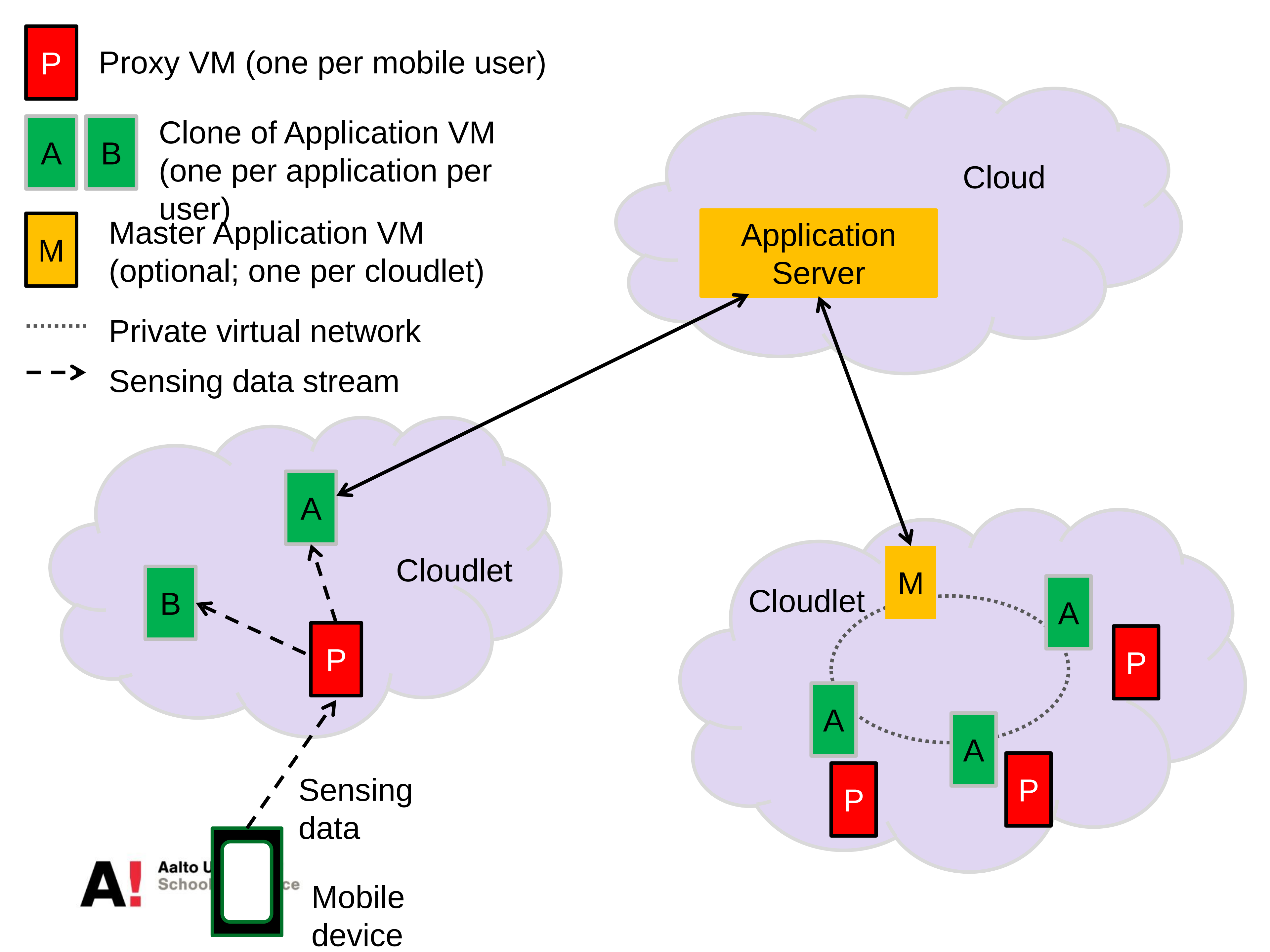
  A cloudlet can be viewed as a "data center in a box" whose goal is to "bring the cloud closer“

  [2] Mahadev Satyanarayanan, Paramvir Bahl, Ramón Caceres, and Nigel Davies. 2009. The Case for VM-Based Cloudlets in Mobile Computing. *IEEE Pervasive Computing* 8, 4 (October 2009), 14-23.

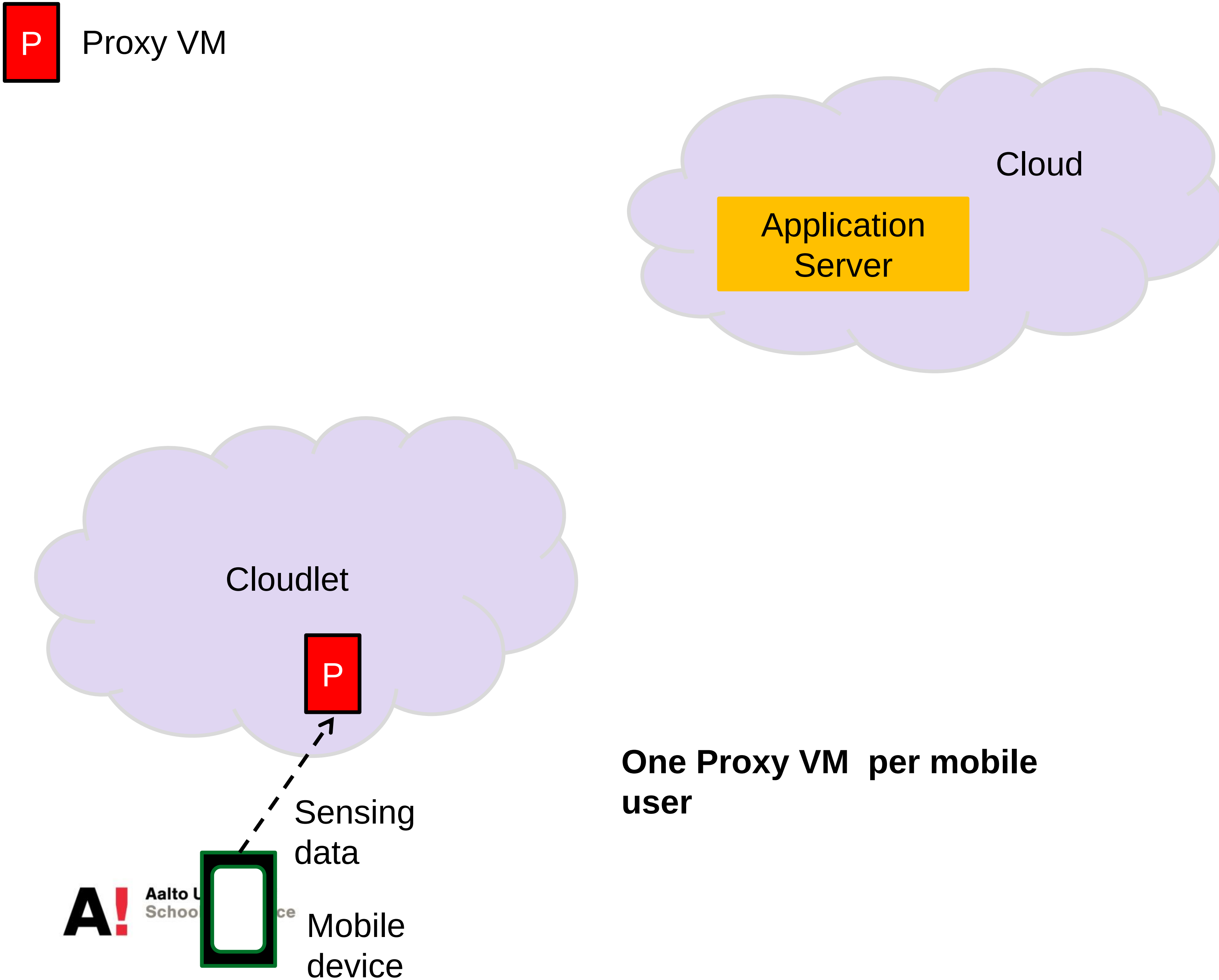
- An opensourced cloudlet framework is being developed by Carnegie Mellon University
  - The framework is built on standard cloud technology
  - Application servers can be installed on the virtual machines that run on distributed cloudlets

- NSN Liquid Application

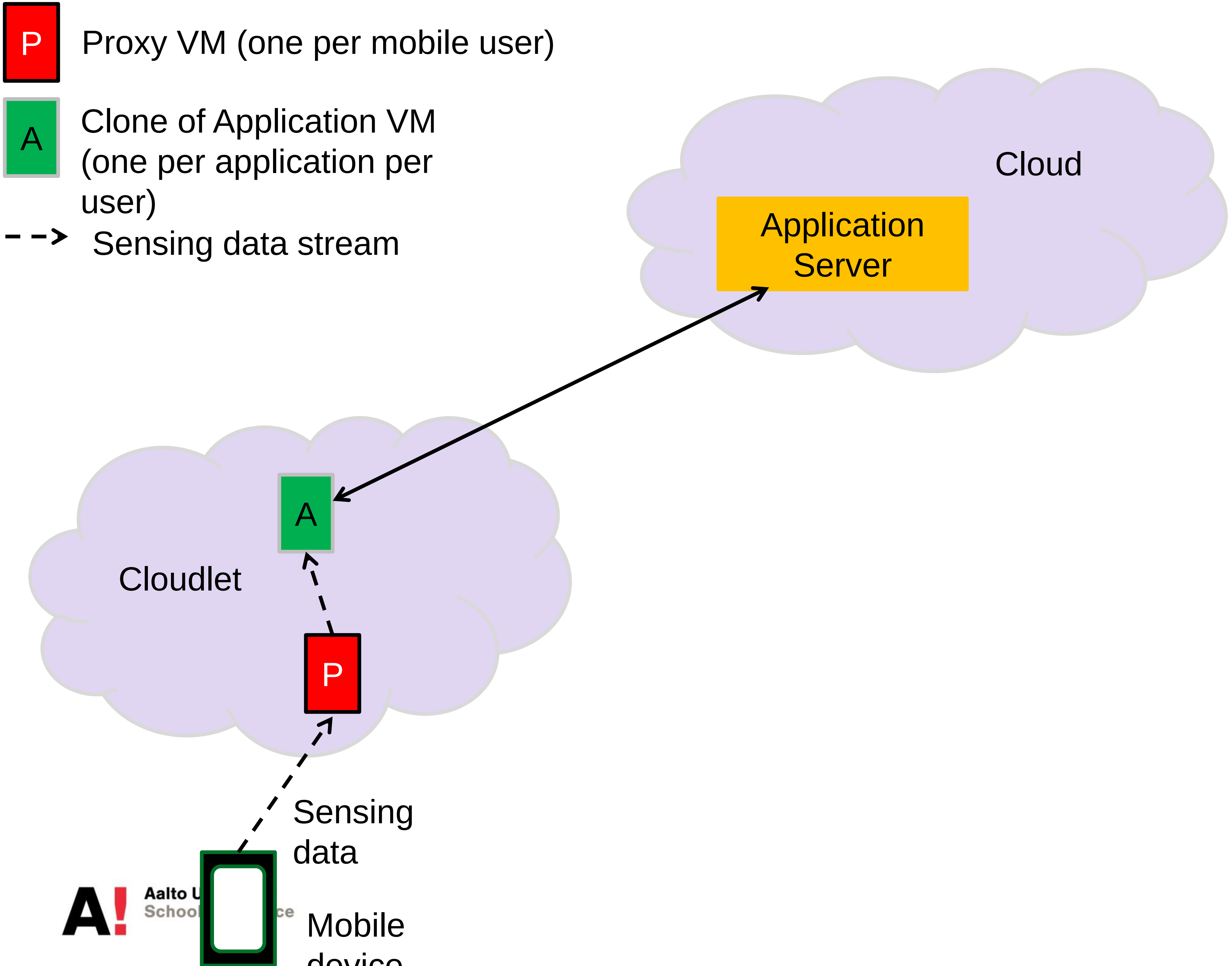
http://nsn.com/sites/default/files/document/liquid-apps-video-edited_0.mp4

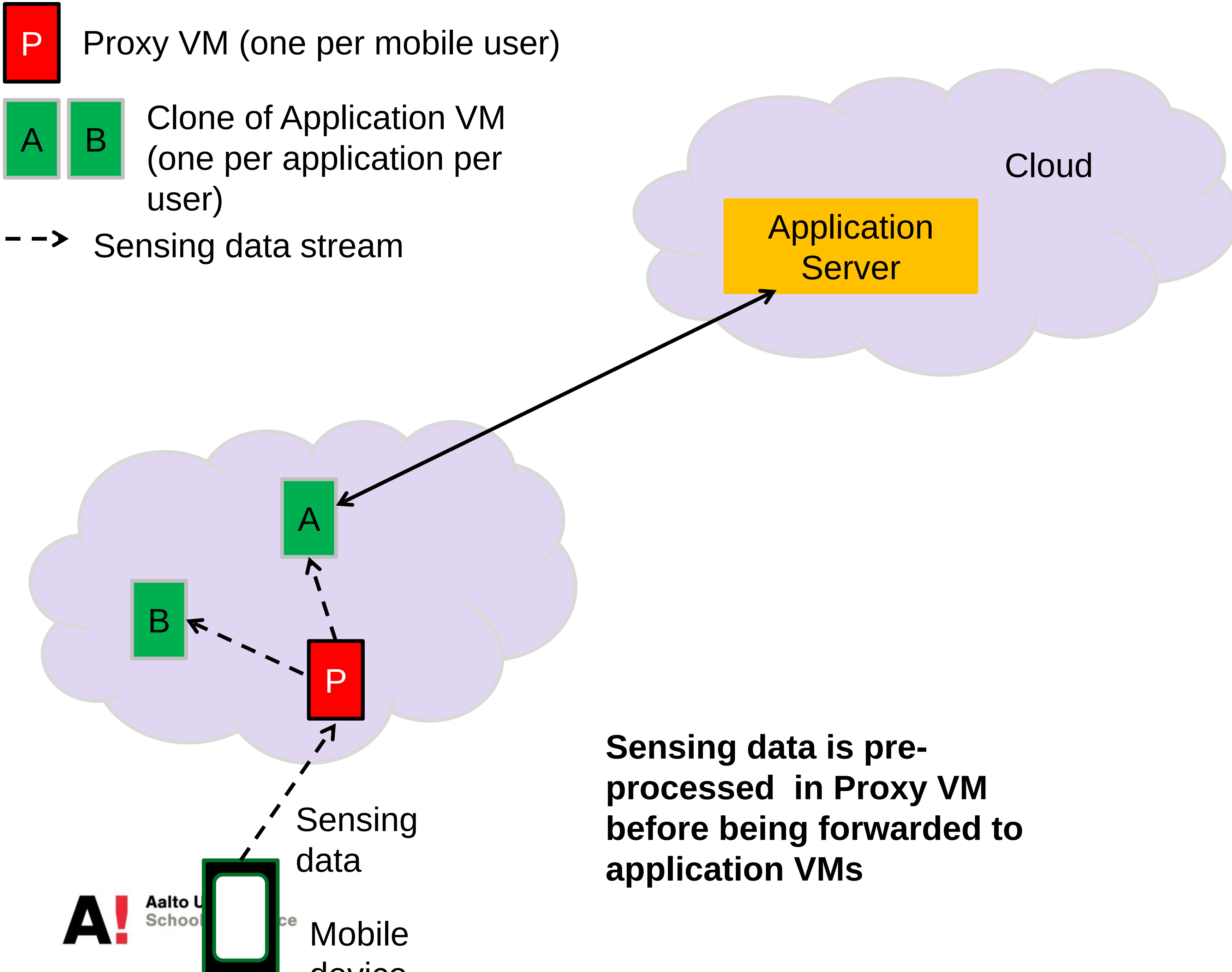


P
Proxy VM (one per mobile user)
A
B
Clone of Application VM (one per application per user)
M
Master Application VM (optional; one per cloudlet)
Private virtual network
Sensing data stream
Cloud
Application Server
Cloudlet
A
B
P
Sensing data
Mobile device
Cloudlet
M
A
A
A
P
P
P

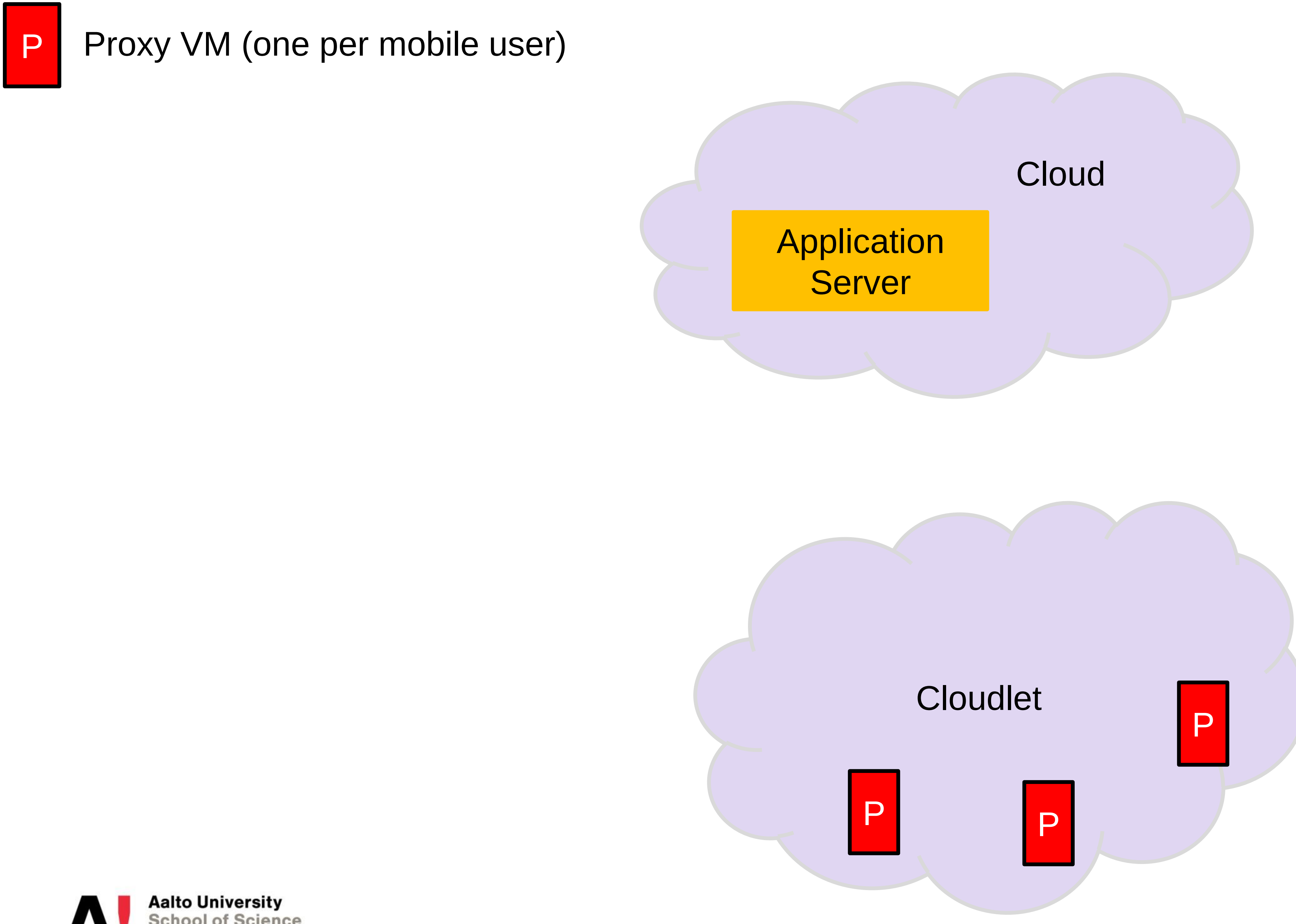


One Proxy VM per mobile user


P
Proxy VM (one per mobile user)
A
Clone of Application VM (one per application per user)
Sensing data stream
Cloud
Application Server
A
Cloudlet
P
Sensing data
Mobile device



Sensing data is pre-processed in Proxy VM before being forwarded to application VMs


P
Proxy VM (one per mobile user)
Cloud
Application Server
Cloudlet
P
P
P

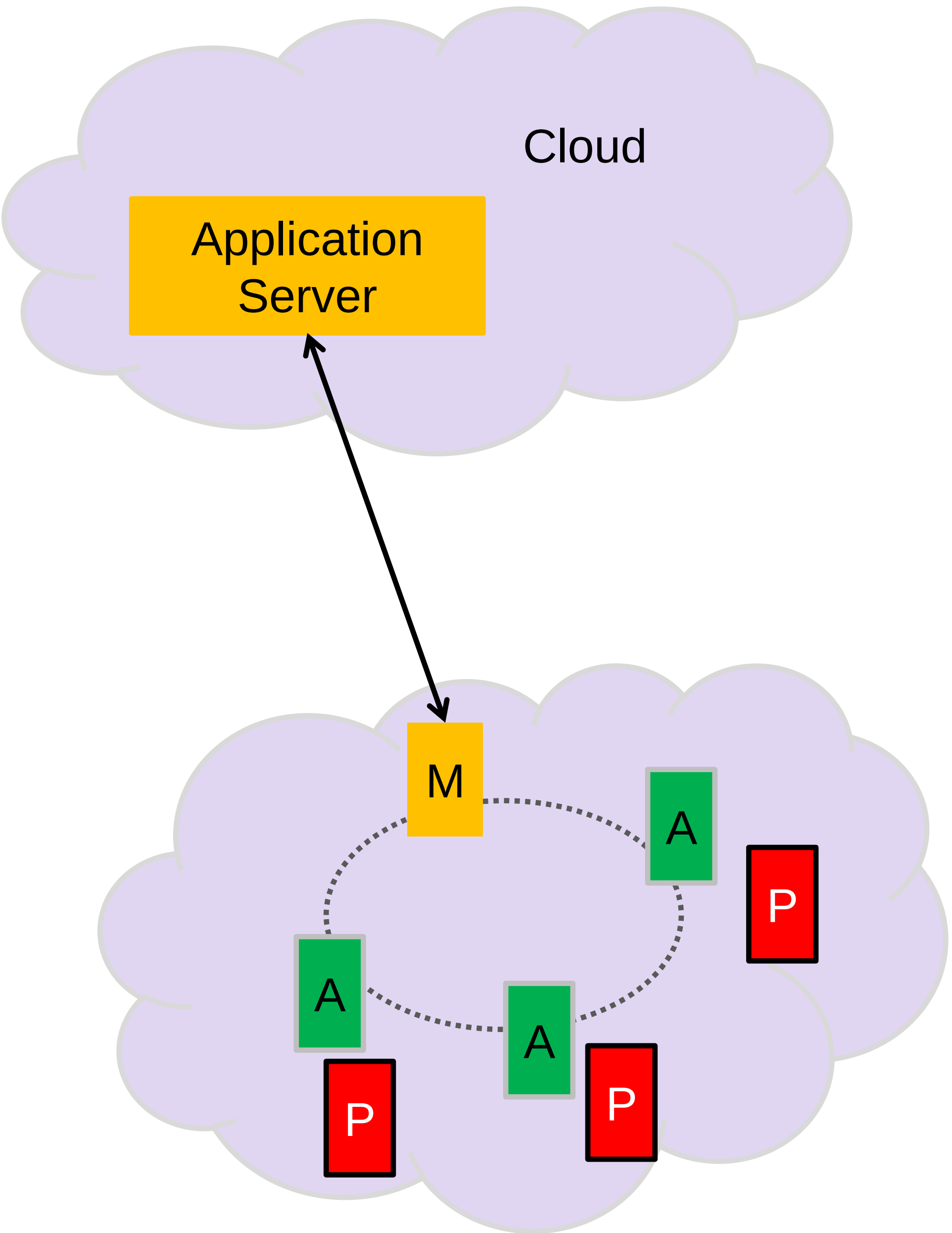
P Proxy VM (one per mobile user)

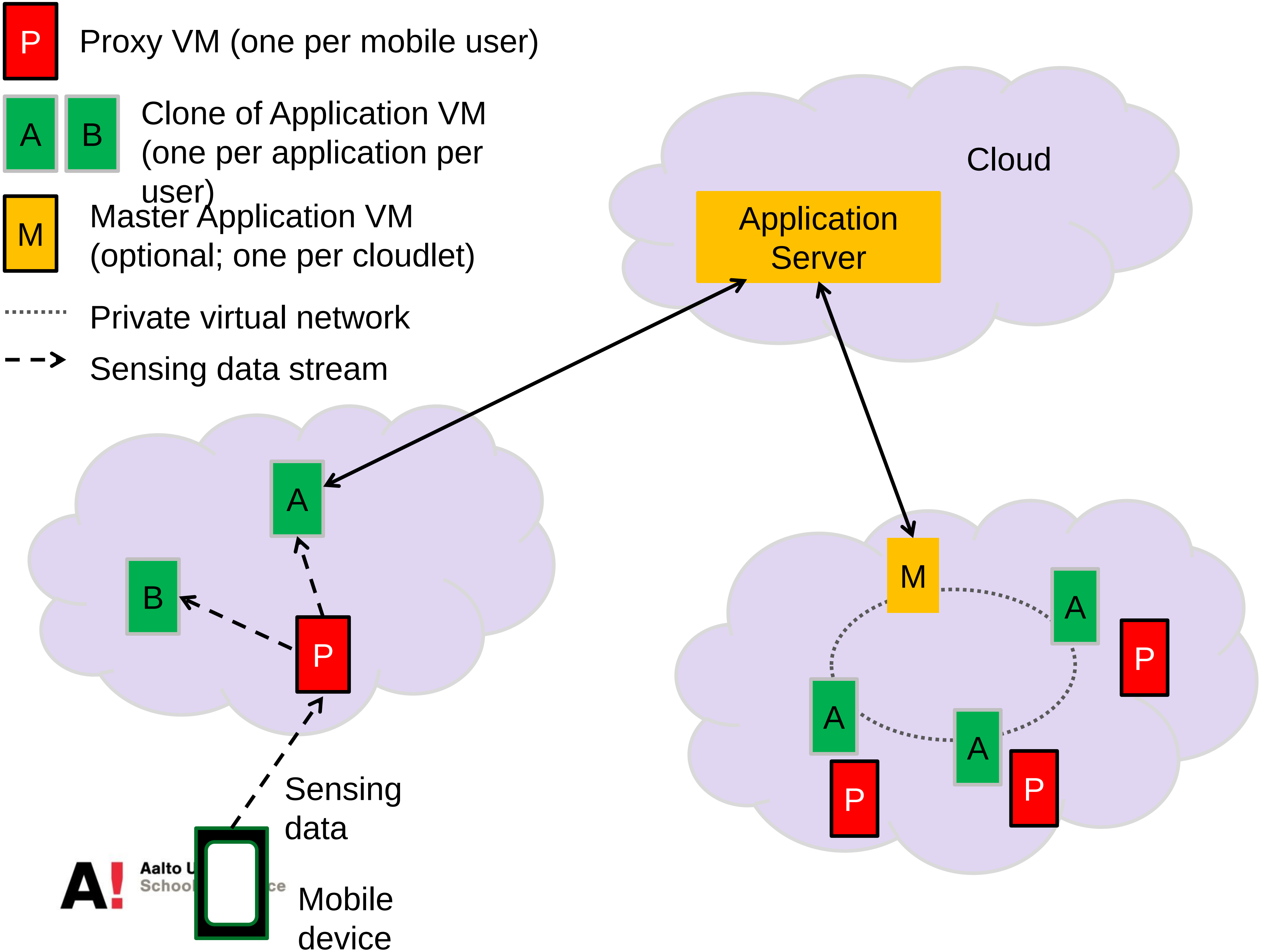
A Clone of Application VM (one per application per user)

M Master Application VM (optional; one per cloudlet)

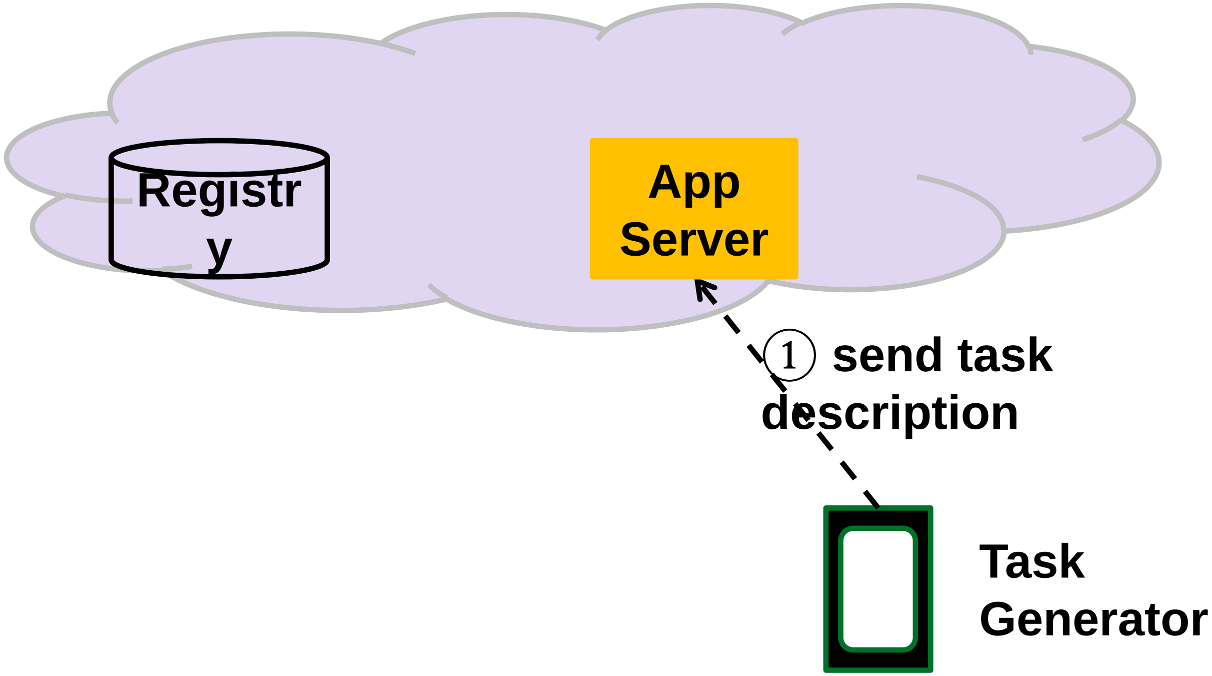
Private virtual network

Sensing data stream

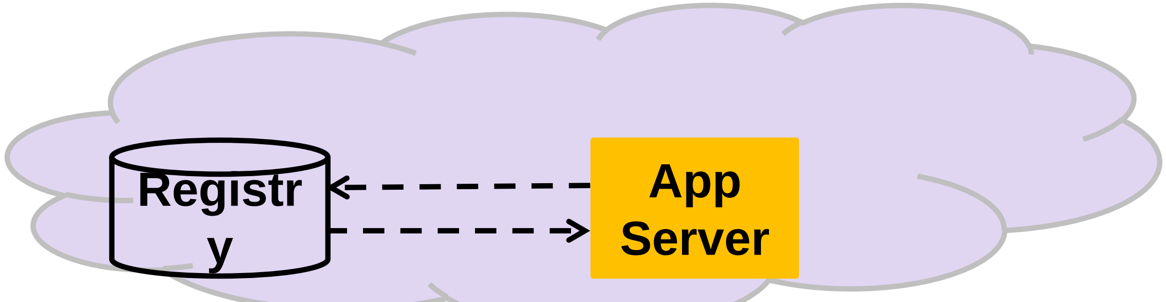



P
Proxy VM (one per mobile user)
A
B
Clone of Application VM (one per application per user)
M
Master Application VM (optional; one per cloudlet)
Private virtual network
Sensing data stream
Cloud
Application Server
A
B
P
Sensing data
Mobile device
M
A
P
A
A
P
P

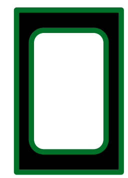
Case study: Finding a lost child from a crowd



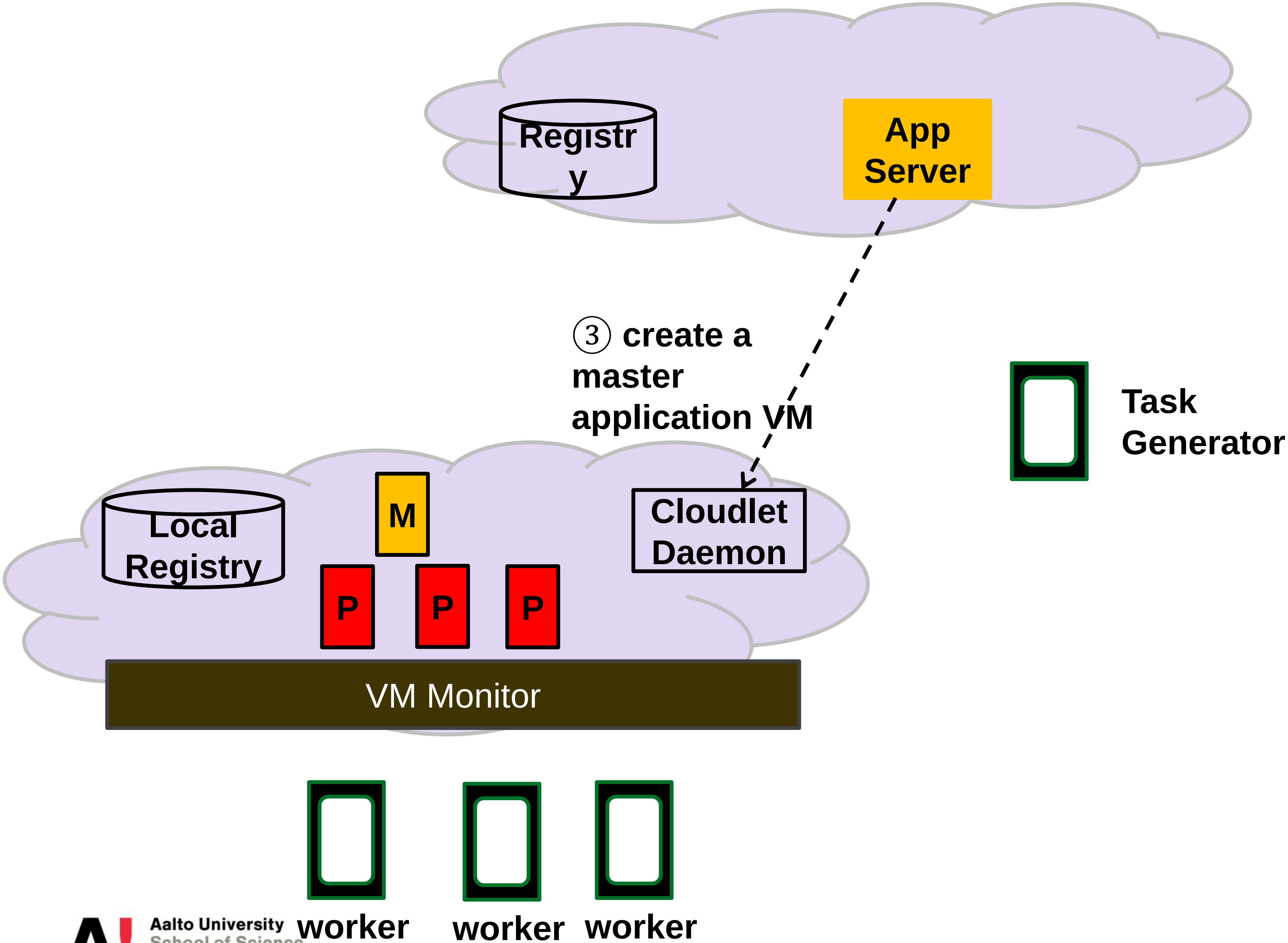
Example Task Description:
<location> Fifth Avenue , New York</location>
<time> 13.00-13.30EST 30.1.2013</time>
<action> face detection </action>
<output> GPS </output>
<attachments>
<image>photo_1.img</image>
<image>photo_2.img</image>
</attachments>



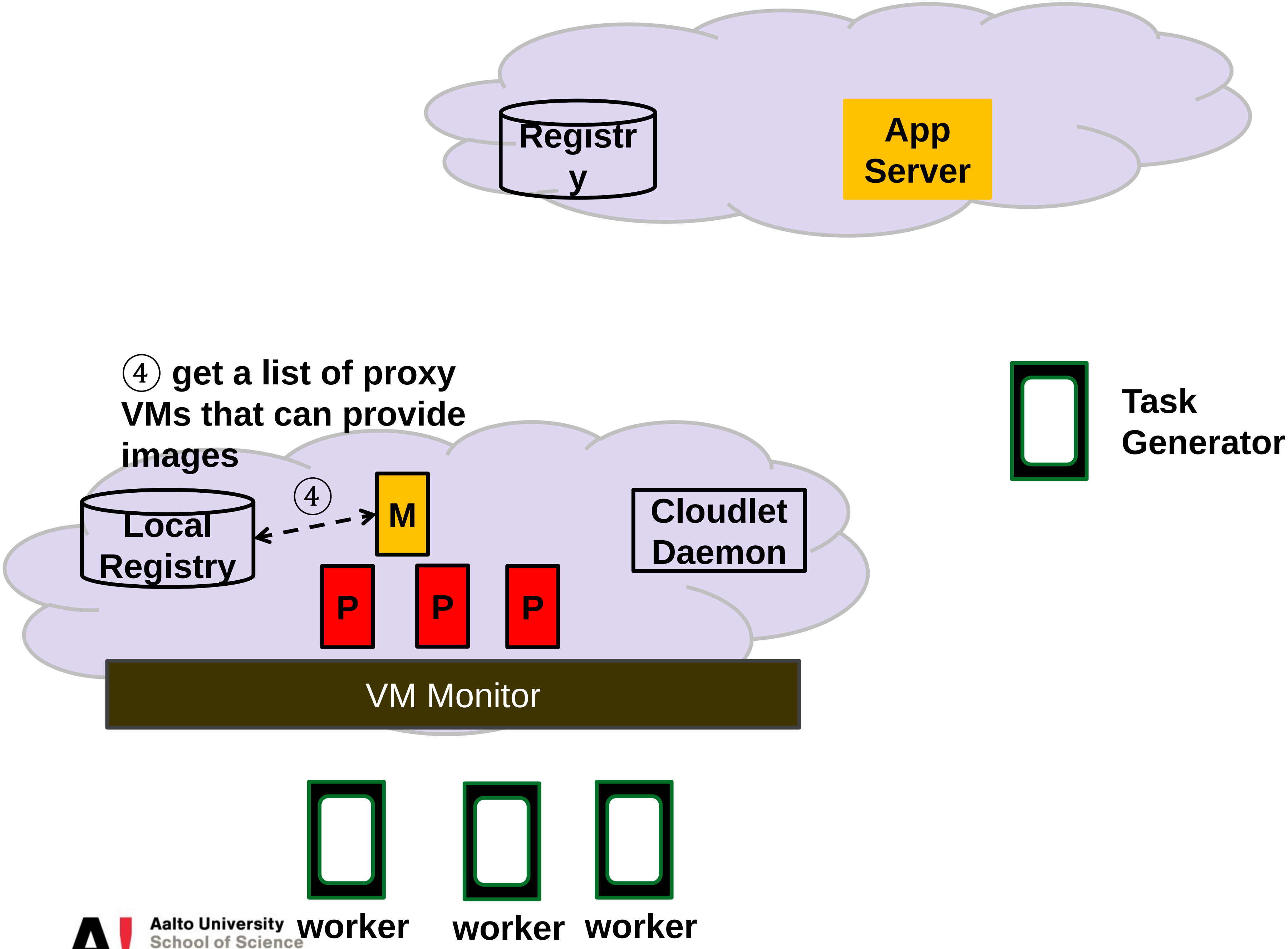
② **get a list of cloudlets located in the target area**



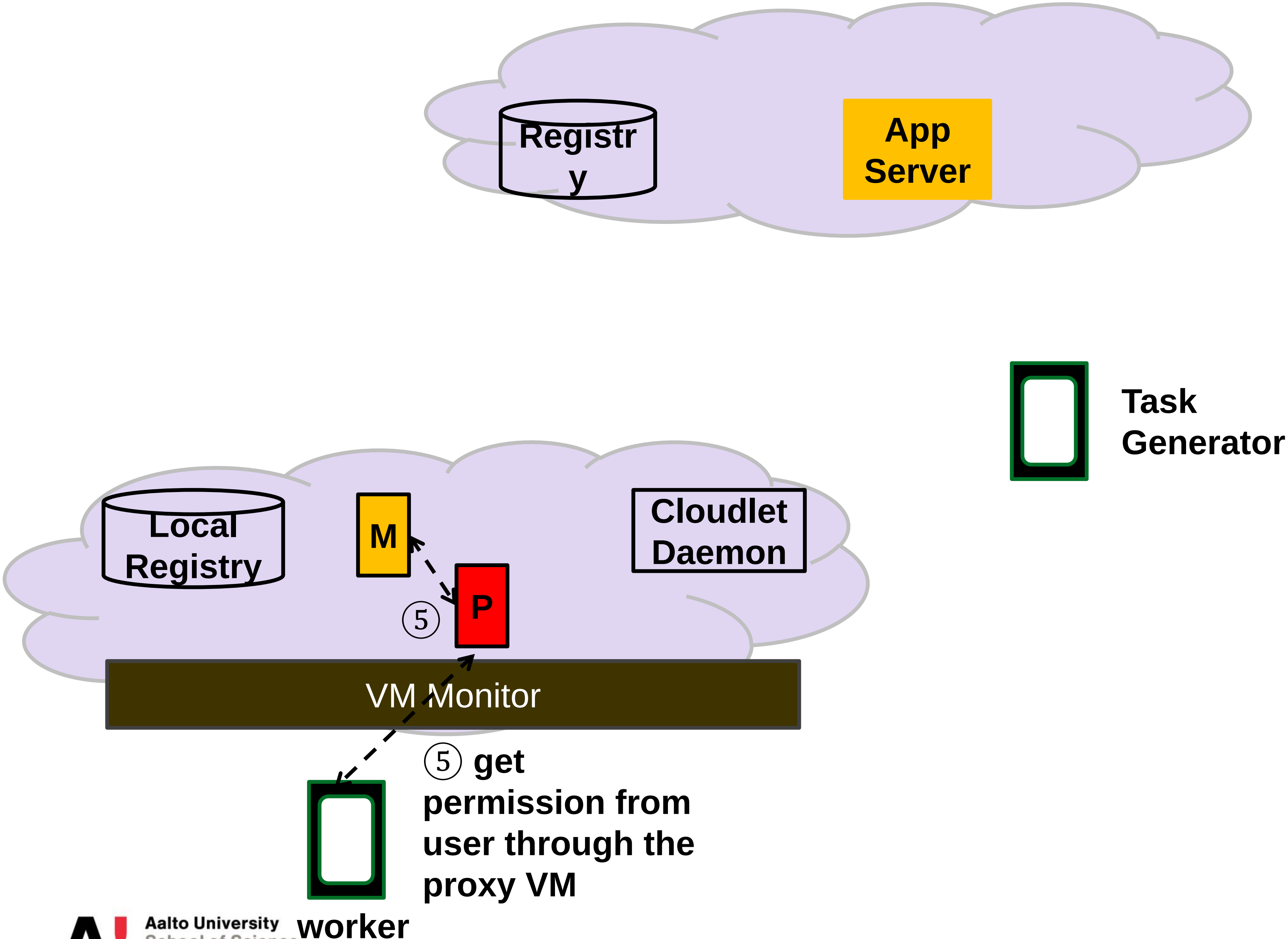
Task Generator

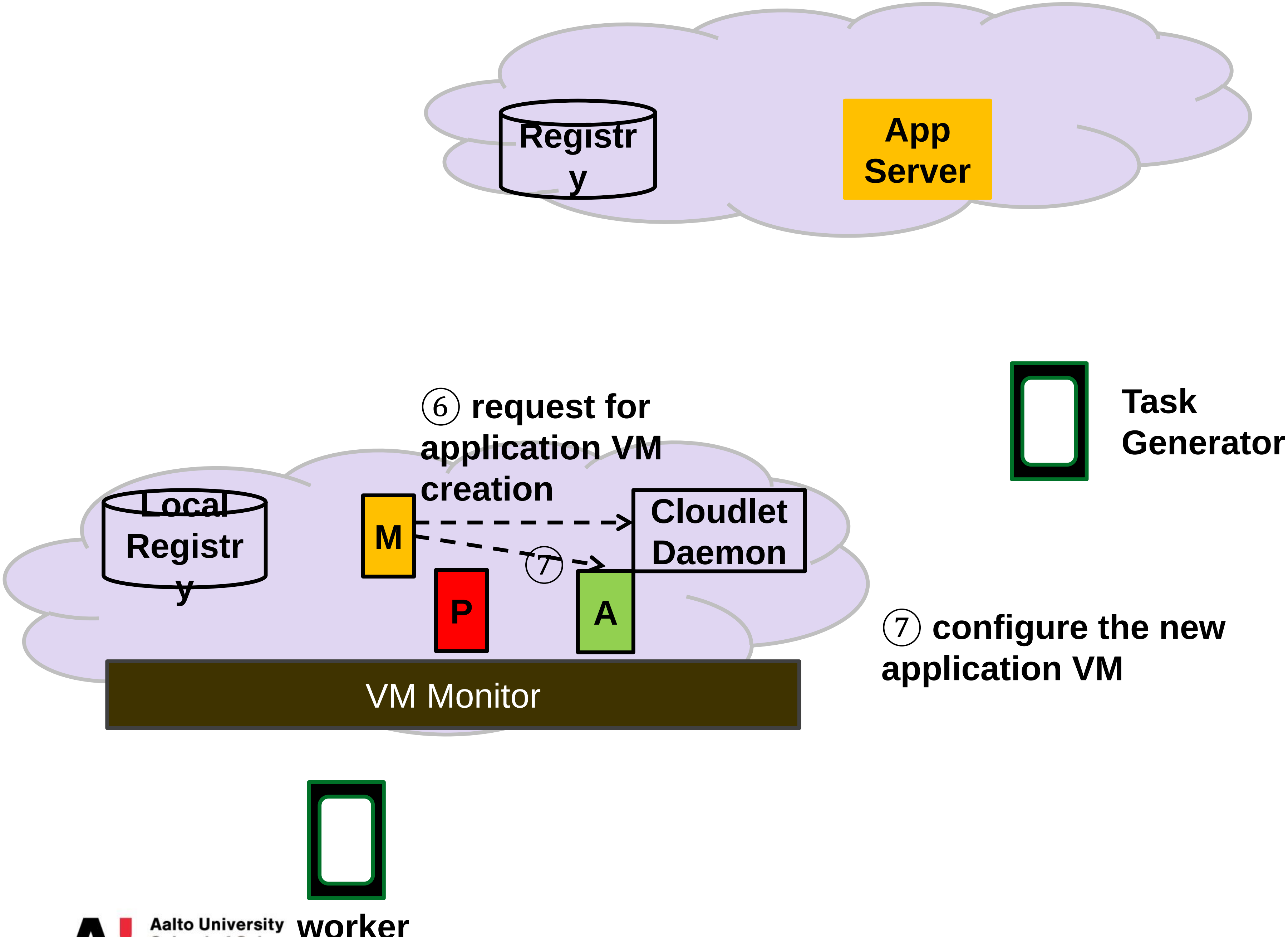


Aalto University
School of Science

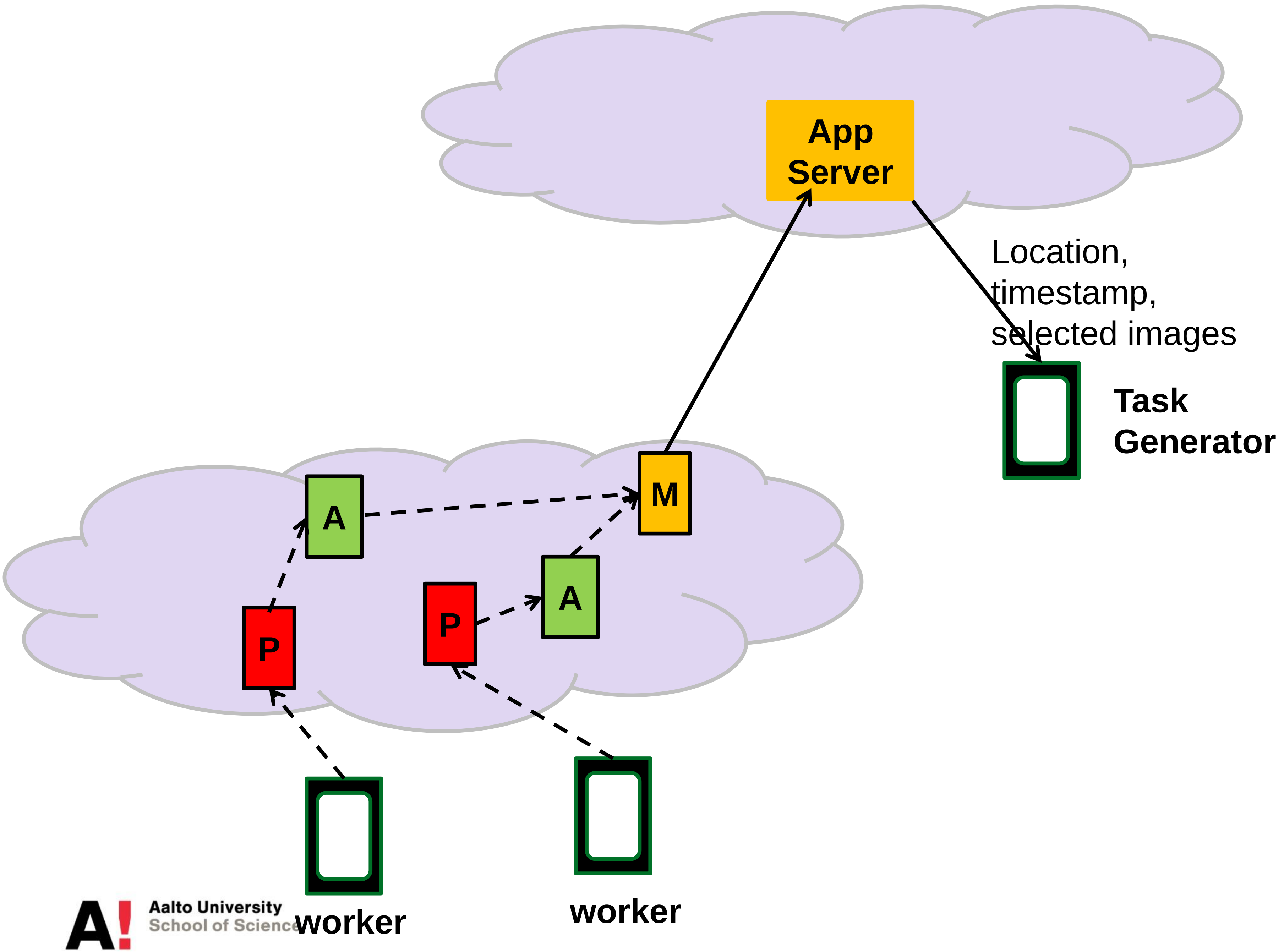


Aalto University
School of Science


Registr
y
App
Server
Task
Generator
Local
Registry
M
Cloudlet
Daemon
⑤
P
VM Monitor
⑤ get
permission from
user through the
proxy VM
worker



Registr
y
App
Server
Task
Generator
⑥ request for
application VM
creation
Local
Registr
y
M
Cloudlet
Daemon
⑦
P
A
⑦ configure the new
application VM
VM Monitor
worker


App
Server
Location,
timestamp,
selected images
Task
Generator
M
A
A
P
P
worker
worker

Deployment Model of CrowdSensing Apps in Ubiqutious Cloud Environment

- Separation of data collection and sharing from application-specific logic

- Removal of installation from the critical path of application deployment

- Decentralization of processing and data aggregation near the source

Challenges

- Virtualization overhead
- Migration-induced reconfiguration
- Standardization of sensing interfaces

Challenges

- Virtualization overhead
- Migration-induced reconfiguration
  - Migration of potentially stateful proxy and application VMs
  - Network reconfiguration

Challenges

- Virtualization overhead
- Migration-induced reconfiguration
- Standardization of sensing interfaces

What if apps require same sensor data but with different format /sampling rate?

Summary

- Sensors in Mobile Consumer Devices
- Personal and CrowdSensing Applications
- Barriers to Large-scale Crowdsensing
- Deployment Model of Crowdsensing Applications in Ubiquitious Cloud Environment