

Team
B!g_Data_Army
Table #2

Problem Statement #2

Internet of Things

8424

Number of Reported Burglaries *

322

Reported Burglaries per 100,000 Residents*


15

% Increase from 2018 to 2019**

* Toronto Police Services stats

** Statistics Canada

**Existing alarm systems are more
reactive than proactive**

Several thin, white, parallel diagonal lines are positioned in the bottom right corner of the slide, extending from the right edge towards the center.

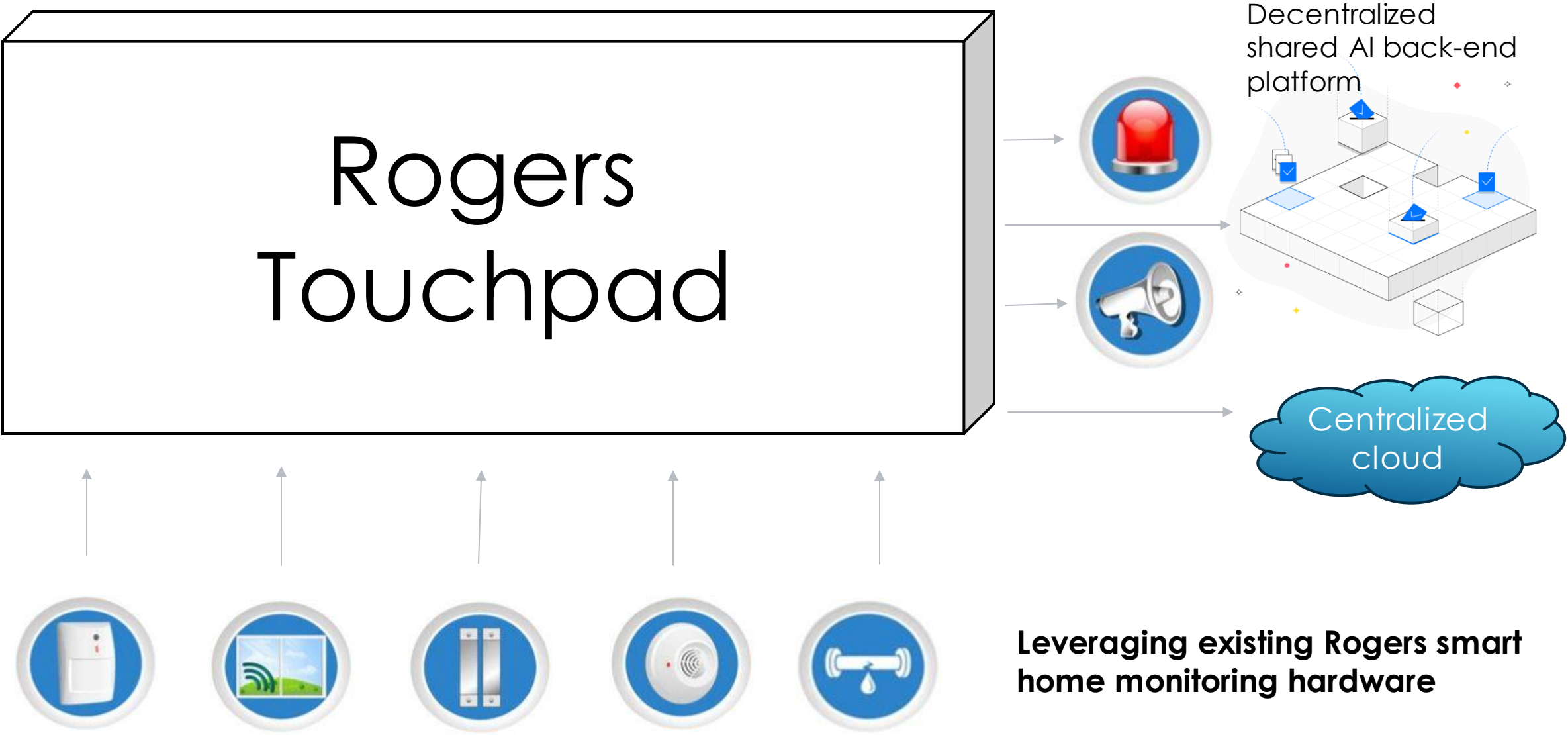
Crowd Security

AI Powered Behaviour Detection, Prediction and Prevention System

Shared Computing Platform Leveraging the Synergy of Communication Network

Three parallel white lines of varying lengths and slopes are positioned in the bottom right corner of the slide, extending from the right edge towards the center.

Solution Architecture



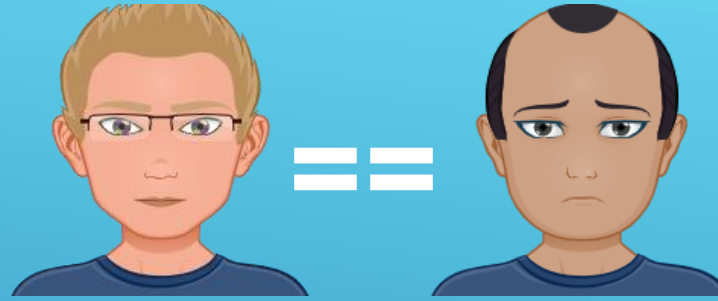
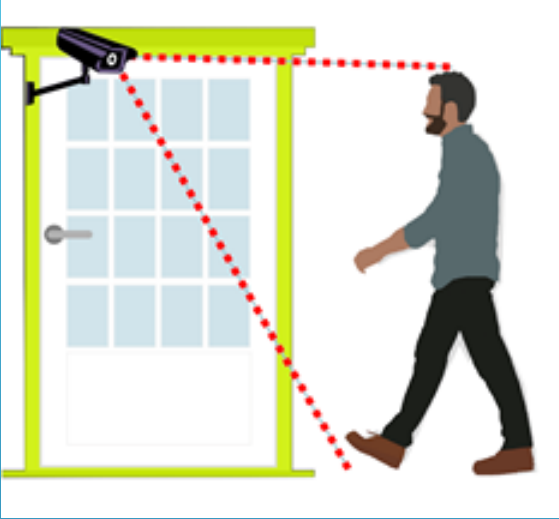
AI models

- Voice Matching
- Face Matching
 - Same face on Backyard
 - Same face on Neighborhood
- Mask Detection
- Weapon Detection
- Door-knob Attempt
- Car-handle Attempt
- Motion Detection
- Electronic Signatures
- Broken Glass Sound Pattern
- Broken Door Sound Pattern
- Gunshot Sound Pattern
- Profile builder
- Low risk profile
 - Flyer delivery
 - Postman
 - Lawn care

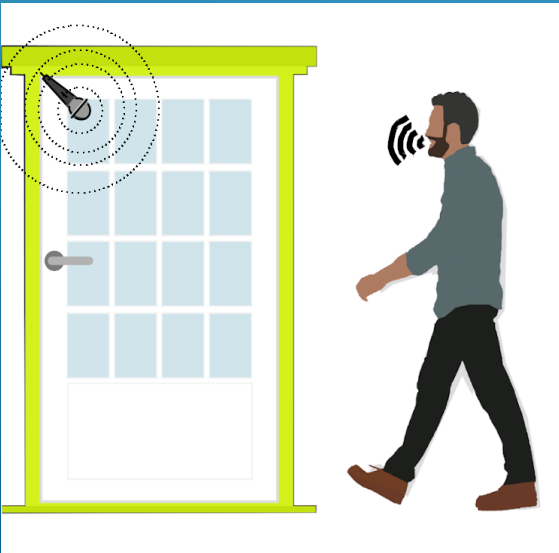
Recognize Patterns

Preventive Actions

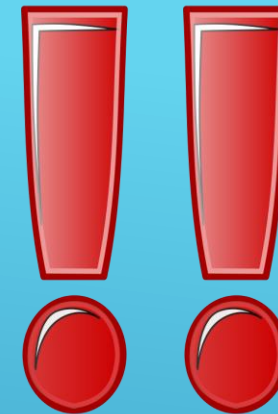
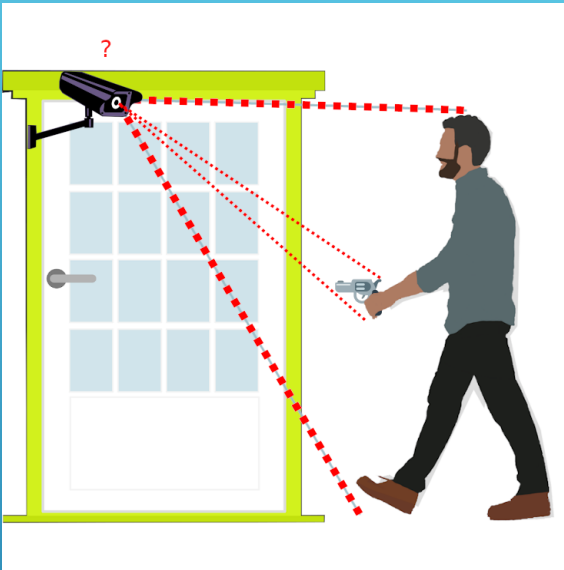
- Radio On/Off
- TV On/Off
- Outdoor lights
- Indoor Lights
- Alarm Siren
- Strobe Lights
- Escalation to Security Company



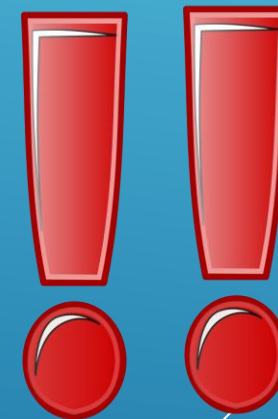
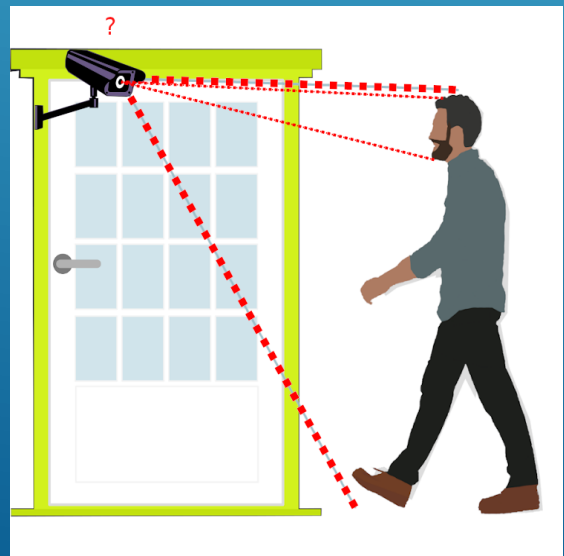
AI-powered image processing compares the face of the front-door person with the ones of the residents stored in the system. Decide if YES or NO



AI-powered sound processing compares the voice of the front-door person with the ones of the residents stored in the system. Decide if YES or NO



AI-powered image processing detects if the front-door person has weapons or robbery tools. Decide if YES or NO



AI-powered image processing detects if the person in front of the door has his face covered with a mask. Decide if YES or NO

Examples of Event Corellation

- Person presence detected
- Face matching
- Voice matching
- Doorbell/knock-on the door
- Door-knob attempt

Event Corellation Algorithm decides on the next preventive actions

- TV Sound on
- Lights on
- Alarm Siren
- Notifying Central Monitoring Station

Different combinations of events in multiple households will adjust the risk profile accordingly

Inter-neighbourhood event corellation captures events occuring in multiple households

Neighbourhood Security Network

- Security events are being transmitted via Neighbourhood Security Network
- The network runs on limited number of neighborhood nodes – vicinity cloud



Crowd Security Participant



- People responding to **security threat events**.
- **Actual people, registered on the platform, and willing to respond**
- Platform app on their cells notifies them of the events
- The algorithm notifies a number of responders, based on proximity
- Duties: drive to location, turn on the high beam lights, honk the horn, observe the surroundings, assess the situation

- If anything suspicious, take pictures, or video record
- On serious assessment, call emergency or police
- Presence gets confirmed by the requesting node, based on the electronic signature of their cell phone and their running app.



Business Value

- **Our platform would be ideal use case for upcoming 5G network in Canada**
- **Increase customer satisfaction**
- **Increase revenue from Services** – monthly-monitoring and some new service offerings
- Increase revenue from Hardware sale
- **Reduction of False Alarms** and disputes/complaints related to False Alarms (Verified Alarm Response regulations being adopted by Police services in North America)
- Reduce headcount at Roger's Central monitoring station
- **Monetization of data collected** – sell to insurance providers and/or used for marketing revenue, research
- Increase reputation of Roger's Home Security System (currently 2 stars)
- Opportunity to grow business beyond residential only, commercial or public spaces, buildings
- Faster response to incidents

Social Benefits

- Customers well-being (Based on Criminal Victimization Study)*
 - 6% have sleeping problems and/or depression
 - 8% have long term PTSD
- Savings for insurance companies and for customers through lower premiums as well as deductibles amount paid via insurance claim

Crowd Security

We put the **Smart** into Smart Home Monitoring

Several thin, white, parallel diagonal lines are positioned in the bottom right corner of the slide, extending from the right edge towards the center.

Q & A



SUPPORTING MATERIALS



Appendix:

•What a Burglar Looks For

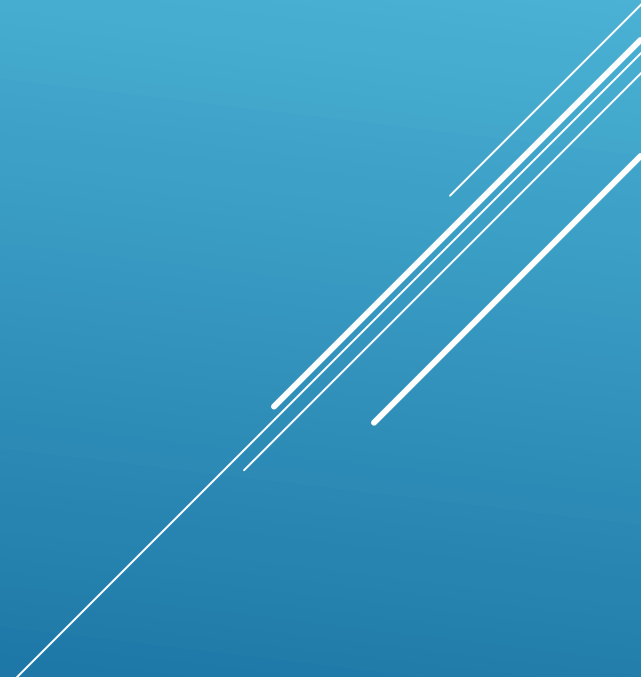
- Most thieves target homes that look easy to break into. They often pick a house by surveilling the neighborhood and finding the one with the most predictable patterns of when people come and go. They also look for properties in more rural or isolated parts of the neighborhood, and houses that are well maintained or have fancy cars in the driveway.
- Most burglars enter houses through those entry points as well as the front door, the back door, or the garage.

•When a Burglar Is Most Likely to Strike

- Burglars want their jobs to be as easy as possible, so if they know you leave for work at 8 a.m. every day and don't return until 6 p.m., they're most likely to strike when you're gone. Summer is the high season for break-ins because many people take summer vacations while their kids are out of school. Many people go on vacations over the major holidays, too, making their homes prime targets for theft.

Appendix:

•**Top 10 deterrents for burglars**

1. CCTV camera
 2. Sound of a barking dog
 3. Strong, heavy doors
 4. TV that has been switched on
 5. Locked UPVC windows
 6. Cars parked on driveway
 7. Overlooking property
 8. Surrounding fences
 9. Gates outside the property
 10. Motion-activated security lights
- 
- A series of three parallel white diagonal lines extending from the bottom right corner towards the center of the slide.

- Appendix:

Electronic Signatures Monitoring

- Nearly all electronic devices, that have wireless connectivity, are broadcasting a continuous and traceable electromagnetic trail – called beacon
- These include: Wi-Fi, Bluetooth, 433MHz, 868MHz and GSM/LTE frequencies
- Examples:
 - Cars – Bluetooth
 - Headsets – Bluetooth, 433Mhz
 - Cell phones – GSM, Wi-Fi, Bluetooth
 - Garage Openers
- These signatures can be captured and recorded as security events

Code snippets for voice matching

```
// voice pattern recognition
import numpy as np
import pandas as pd
import seaborn as sns
sns.set_style('whitegrid')
import matplotlib.pyplot as plt

from keras.models import Model
from keras.layers import Dense, Dropout, Bidirectional, CuDNNGRU, Reshape,
GlobalMaxPooling1D, GlobalAveragePooling1D, Input, concatenate,
BatchNormalization
from keras.optimizers import Adam
from keras.callbacks import ReduceLROnPlateau
from sklearn.metrics import accuracy_score

// Using TensorFlow backend.
```

Code snippets for Object Recognition (Weapons)

Object detection model using YOLO Darknet to detect harmful weapons such as gun and knife, in the hands of a person.

```
def detect(net, meta, image, thresh=.5,
hier_thresh=.5, nms=.45):
im = load_image(image, 0, 0)
boxes = make_boxes(net)
probs = make_probs(net)
num = num_boxes(net)
network_detect(net, im, thresh, hier_thresh, nms,
boxes, probs)
res = []
for j in range(num):
for i in range(meta.classes):
if probs[j][i] > 0: res.append((meta.names[i],
probs[j][i], (boxes[j].x, boxes[j].y, boxes[j].w,
boxes[j].h)))
res = sorted(res, key=lambda x: -x[1])
free_image(im)
free_ptrs(cast(probs, POINTER(c_void_p)), num)
return res
```



Code snippets for Electro-Magnetic Device Signature Monitoring (Wifi)

<https://github.com/kissste/esp8266-Arduino-WifiSniffer/blob/master/ESP8266-Arduino-WifiSniffer.ino>

```
void setup () {  
  Serial.begin(115200); delayMicroseconds(100);  
  Serial.println("*** Monitor mode test ***"); Serial.print(" -> Promisc mode setup ... ");  
  
  wifi_set_promiscuous_rx_cb(promisc_cb);  
  wifi_promiscuous_enable(1);  
  Serial.println("done.");  
  Serial.print(" -> Timer setup ... ");  
  os_timer_disarm(&channelHop_timer);  
  os_timer_setfn(&channelHop_timer, (os_timer_func_t*) channelHop, NULL);  
  os_timer_arm(&channelHop_timer, CHANNEL_HOP_INTERVAL, 1);  
  Serial.println("done.\n");  
  Serial.print(" -> Set opmode ... ");  
  wifi_set_opmode(0x1 );  
  Serial.println("done.");  
  Serial.println(" -> Init finished!");  
}  
  
void loop() { delay(10); }
```

Code snippets for Electro-Magnetic Device Signature Monitoring (Bluetooth - BLE)

https://github.com/moononournation/Arduino_BLE_Scanner/blob/master/Arduino_BLE_Scanner.ino

```
BLEDevice::init("");


// put your main code here, to run repeatedly:
BLEScan *pBLEScan = BLEDevice::getScan(); //create new scan
pBLEScan->setAdvertisedDeviceCallbacks(new MyAdvertisedDeviceCallbacks());
pBLEScan->setActiveScan(true); //active scan uses more power, but get results faster
pBLEScan->setInterval(0x50);
pBLEScan->setWindow(0x30);

#ifdef SERIAL_PRINT
Serial.printf("Start BLE scan for %d seconds...\n", SCAN_TIME);
#endif

BLEScanResults foundDevices = pBLEScan->start(SCAN_TIME);

for (int i = 0; i < count; i++)
{
    BLEAdvertisedDevice d = foundDevices.getDevice(i);
    if (d.haveManufacturerData())
    {
        std::string md = d.getManufacturerData();
        uint8_t* mdp = (uint8_t*)d.getManufacturerData().data();
        char *pHex = BLEUtils::buildHexData(nullptr, mdp, md.length());
        ss << ",\nManufacturerData\":" << pHex << "\n";
        free(pHex);
    }
}
```

AI based Security Threat Risk Assessment

- Events from vicinity nodes are fed to Machine Learning (ML) algorithm which predicts risk of threat
 - As a result of the risk assessment, deterrents might be activated.
 - Deterrents can be local or attached to the vicinity nodes (For example – neighbor's house front door lights would turn on)
 - Risk threat level and the distance to the source of events will influence nodes participation (deterrent as well as notification of participants and residents).
- 
- A series of four parallel white diagonal lines of varying lengths, located in the bottom right corner of the slide, pointing towards the top right.

Back-end Platform Concepts

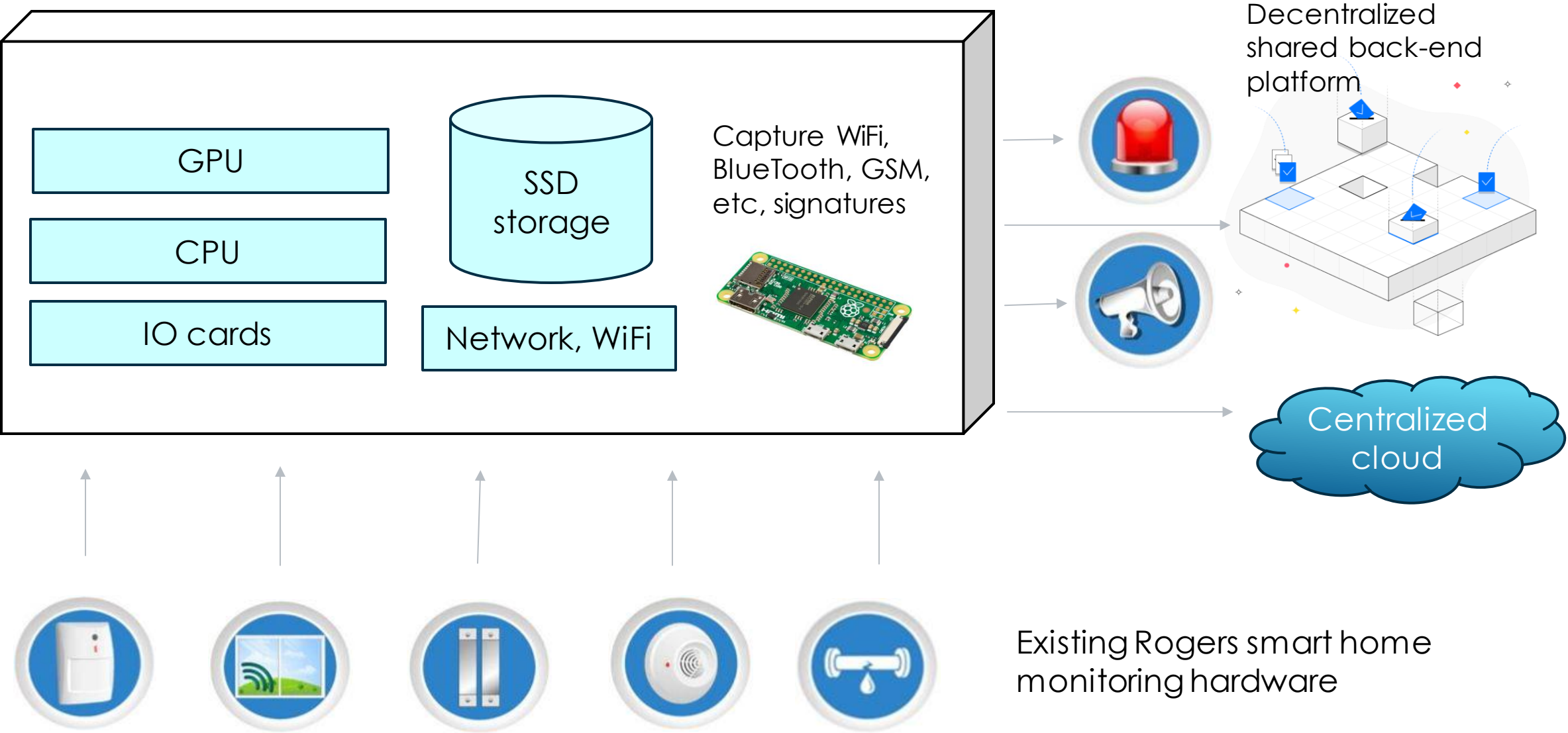
- The back-end platform is a **Distributed Ledger Technology** (DLT) which is a blockchain, running on limited number of neighborhood nodes – **vicinity clouds**
- A blockchain of **security events**
- **Vicinity clouds** – no actual borders, they overlap, mix and continuously cross over each other



Back-end Platform Concepts - cont

- The blockchain records **security events**, each one only within its own vicinity cloud
- Basic **consensus**, based on replication of the security events
- To get accepted on the blockchain, each node needs **proof-of-stake**, a minimum amount of the platform's crypto-coin in its wallet.
- The master nodes perform **proof-of-computing** to maintain the DL, by performing AI image processing for the validation nodes
- Validator nodes confirm the proof-of-computing, as events on the blockchain.
- The master nodes are rewarded with the **crypto-coin of the platform**, for performing the proof-of-computing
- Each unit of computing done by a master node for a validation node gets paid from the **wallet of the validation node**
- Owners of any participating node can sign for the **crowd security sourcing**, and respond to a serious security event, or **security threat**.
- Each **security action response** of a **crowd security participant**, as a result of a security threat, gets rewarded with the crypto-coin of the platform, from the wallet of the node requiring the attention or intervention.
- **The is NO crypto-coin mining**; all the coins consumed by the proof-of-computing or security action responses, are sourced from the wallets of the validation nodes, or from the wallets of nodes requiring intervention.

Unit Hardware Architecture



Unit Software Architecture



AI models

Voice Matching	Electronic Signatures
Face Matching	Same face on Backyard
Mask Detection	Same face on Neighborhood
Weapon Detection	Profile builder
Door-knob Attempt	Broken Glass Sound Pattern
Car-handle Attempt	Broken Door Sound Pattern
Motion Detection	Gun-shot Sound Pattern
Recognize Event Patterns	

User Interface
- setup
- configuration

**Algorithms, decision,
deterrents/ actuators**

Database

File system

Blockchain wallet

**Blockchain validator and/or
full master node**

**Graphics
Shared-processing
computing module**

**Cloud comm and
Secured API end-points**

Audio drivers

Video streaming drivers

EM sniffing drivers

Motion and other drivers

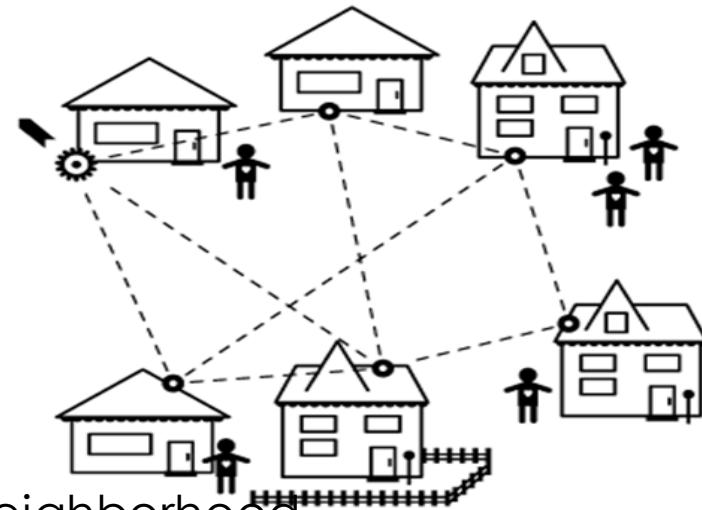
OS (Operating System)

Back-end Platform Architecture



Master node (the equivalent of a miner node)

- Runs the master node software of the blockchain platform
- Has one or more high-end graphics cards
- Maintains the distributed ledger of events for the units in the neighborhood
- Provides image processing for the assigned validation nodes
- Gets rewarded with the crypto-coin of the network for the proof-of-computing
- Does not get rewarded for maintaining and replicating the distributed ledger
- Sends alarm messages to law enforcement and to the crowd security participants



Validation node

- Runs the validation node software of the blockchain platform
- Maintains the distributed ledger of events for the units in the neighborhood
- Possibly runs the AI models of only its own image processing
- Does not get rewarded for maintaining and replicating the distributed ledger
- Sends alarm messages to law enforcement and to the crowd security participants