# Service Design Workshop

## Mobile Web Services

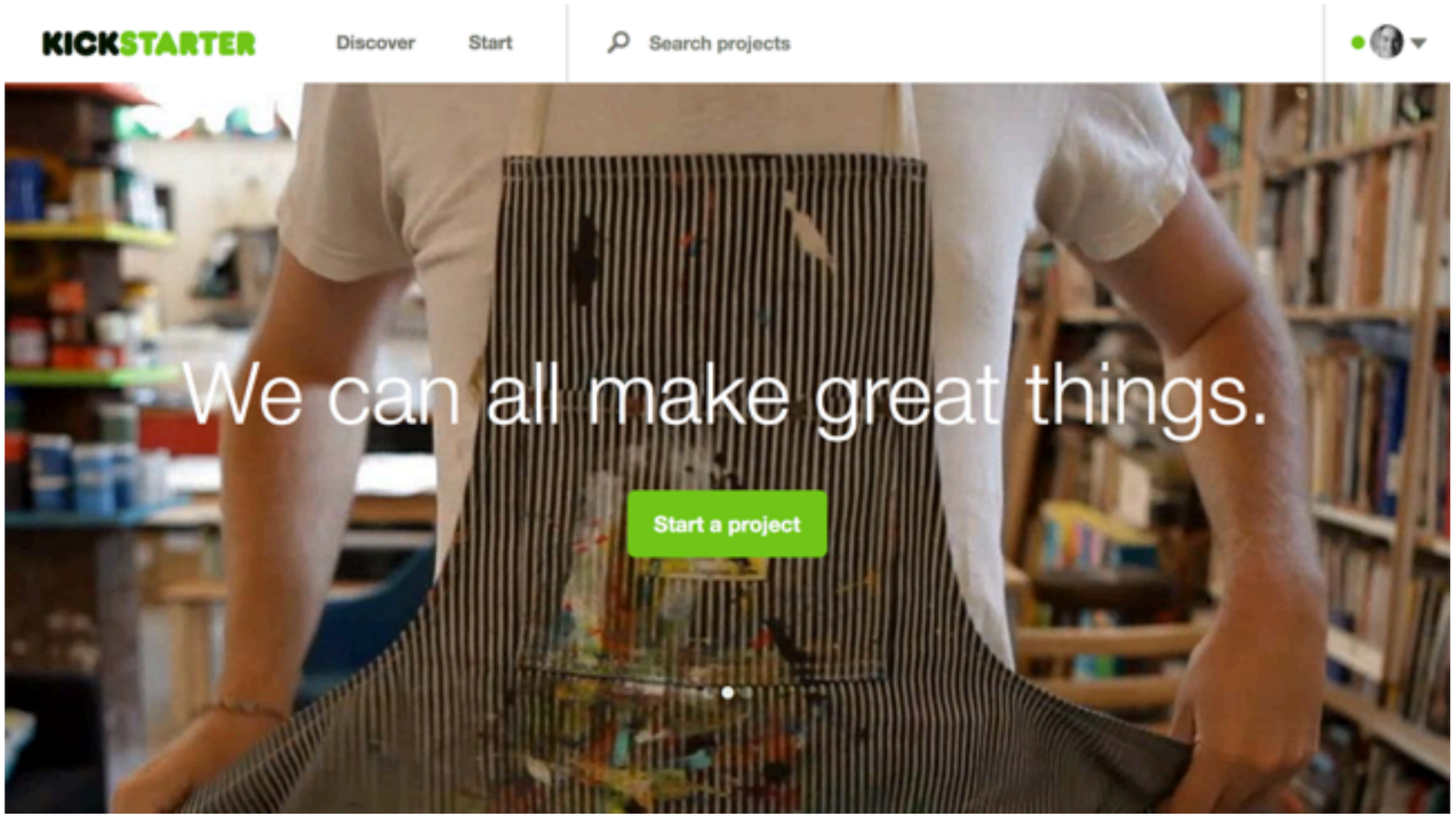
Olivier Liechti & Yannick Iseli

heig-vd

Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

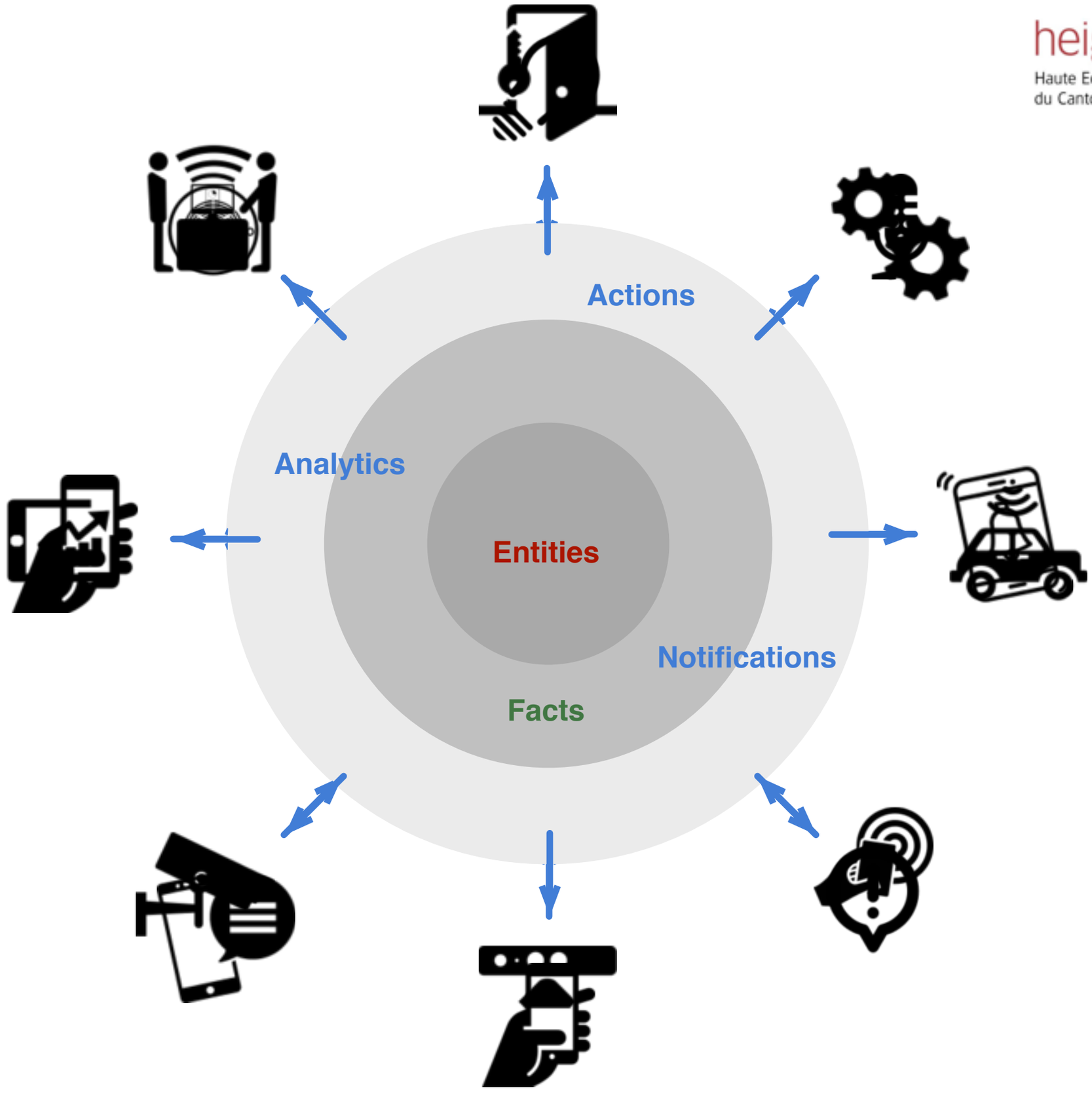

Koubachi
Buy
Learn
Help
Search...
Login
Koubachi analyzes your data
and gives you detailed care instructions
when and how to care for your plants.
2
PCE
Koubachi Plant Care Engine
3
1
Sensor
Apps
Measure
Analyze
Display
1
2
3

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud




**Imagine** and **describe** a **service** that would combine a **mobile app** and some kind of **sensor(s)** for the benefit of a **particular user group**.

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud



heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

"In my **application domain**, there are ***Entities***: plants, bikes, locations, users, buildings, activities, objects, etc."

"By **interpreting** streams of events, I can infer ***Facts*** about domain entities."

"**Sensors** report **streams of *Observations*** about what they see (in the physical or in the digital world)."

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

Knowledge

**Entities**

Locations: Kitchen, Living Room, Basement
People: John, Sarah, Bob, Alice
Meetings: meeting1, meeting2, meeting3

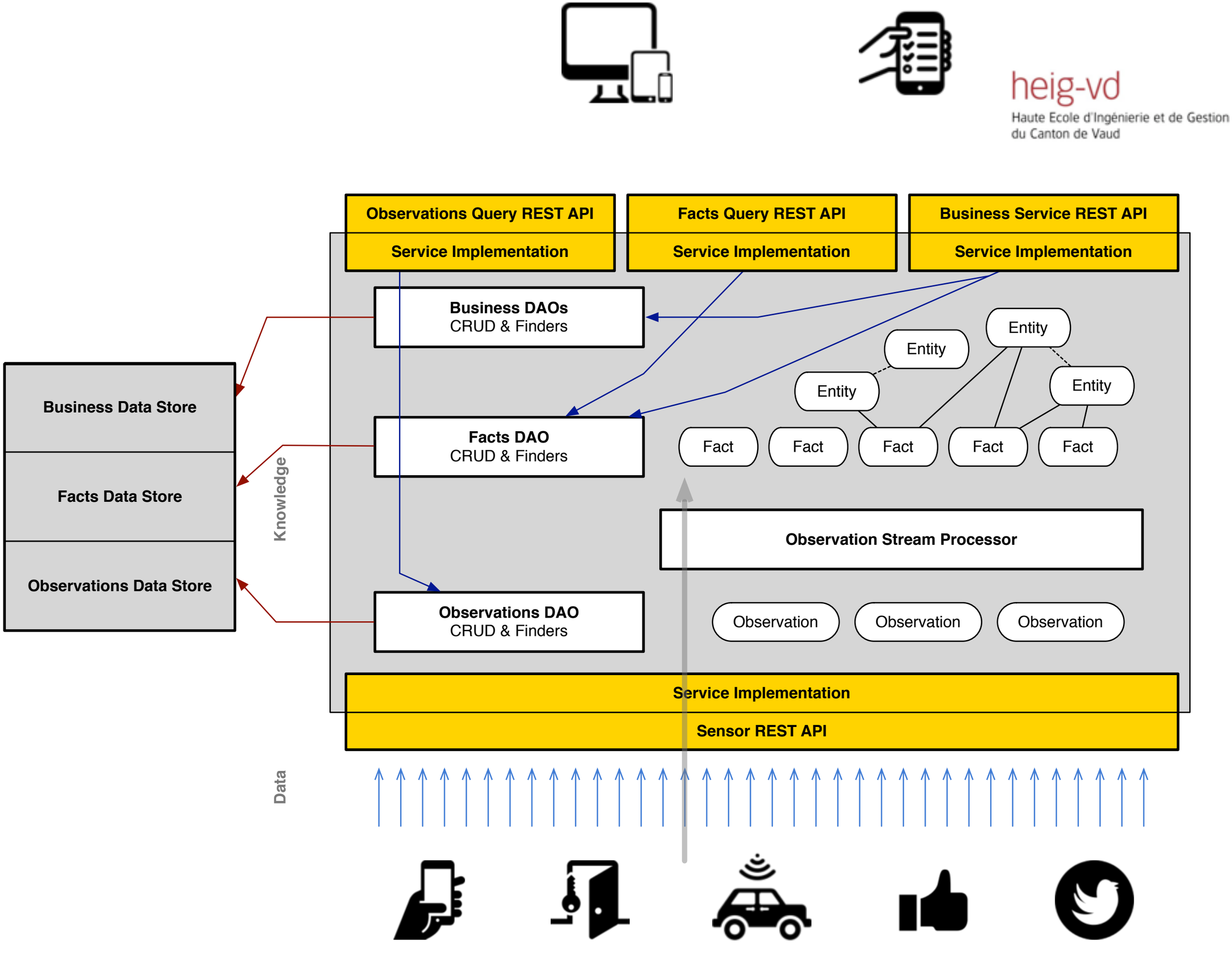
**Facts**

"The last know temperature in the kitchen is of 12.2 degrees (at 12:05:00)"
"The warmest temperature in the kitchen during the last 24 hours is 23.2 d"
"John has been in the elevator at 12:20:00"
"John has been in 5 locations during the last 2 hours"
"John and Sarah have been in three meetings together over last week"

Data

**Observations**

"sensor A2 has measured a temperature of 12.3 degrees at 12:02:00"
"sensor A2 has measured a temperature of 12.2 degrees at 12:05:00"
"sensor A3 has measured a temperature of 12.8 degrees at 12:05:00"
"sensor B8 has seen tag 87KHE8 appear at 12:20:00"
"sensor B8 has seen tag 82UU28 disappear at 12:20:00"
"sensor C6 has sensed a new location at 46°47'25.2"N 6°31'22.1"E"




heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

heig-vd
Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud

**BusinessService REST API**

GET /locations/{id}/periods?filter=warmest
GET /periods/{id}/locations?filter=cool

**BusinessService**

Location findWarmestPeriodForLocation(Location loc)
List<Location> findCoolSummerPlaces()
List<Location> findHotWinterPlaces()

**Configuration REST API**

GET /locations/
GET /locations/{id}
POST /locations
PUT /locations/{id}
DELETE /locations/{id}

GET /periods/
GET /periods/{id}
POST /periods
PUT /periods/{id}
DELETE /periods/{id}

**LocationsDAO**

List<Location> findAll()
Location findLocationById()
void createLocation()
void updateLocation()
void deleteLocation()

**PeriodsDAO**

List<Location> findAll()
Period findPeriodById()
void createPeriod()
void updatePeriod()
void deletePeriod()

**Location**

String name
double latitude
double longitude
double radius

boolean isInLocation(double lat, double lon)

**Period**

long startTime
long endTime

**HighestTemperatureFact**

double temperature

**FactsDAO**

HighestTemperatureFact findFactsByLocation()
HighestTemperatureFact findFactsByPeriod()
createOrUpdateFact(long lat, long lon, long ts, double temperature)

**ObservationsDAO**

List<Observation> findAll()
Location findObservationById()
void createObservation(Observation o)

**TemperatureObservation**

double latitude
double longitude
long timestamp
double temperature

**StreamProcessor**

onObservation(TemperatureObservation o);

**Sensor REST API**

POST /observations/
POST /observations/

**Location**

String name
double latitude
double longitude
double radius

boolean isInLocation(double lat, double lon)

**Period**

long startTime
long endTime

**HighestTemperatureFact**

double temperature

**FactsDAO**

HighestTemperatureFact findFactsByLocation()
HighestTemperatureFact findFactsByPeriod()
createOrUpdateFact(long lat, long lon, long ts, double temperature)

**ObservationsDAO**

List<Observation> findAll()
Location findObservationById()
void createObservation(Observation o)

**TemperatureObservation**

double latitude
double longitude
long timestamp
double temperature

**StreamProcessor**

onObservation(TemperatureObservation o);

**Sensor REST API**

POST /observations/
POST /observations/

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

**BusinessService REST API**

GET /locations/{id}/periods?filter=warmest
GET /periods/{id}/locations?filter=cool

**BusinessService**

Location findWarmestPeriodForLocation(Location loc)
List<Location> findCoolSummerPlaces()
List<Location> findHotWinterPlaces()

**FactsDAO**

HighestTemperatureFact findFactsByLocation()
HighestTemperatureFact findFactsByPeriod()
createOrUpdateFact(long lat, long lon, long ts, double temperature)

**Configuration REST API**

GET /locations/
GET /locations/{id}
POST /locations
PUT /locations/{id}
DELETE /locations/{id}

GET /periods/
GET /periods/{id}
POST /periods
PUT /periods/{id}
DELETE /periods/{id}

**LocationsDAO**

List<Location> findAll()
Location findLocationById()
void createLocation()
void updateLocation()
void deleteLocation()

**PeriodsDAO**

List<Location> findAll()
Period findPeriodById()
void createPeriod()
void updatePeriod()
void deletePeriod()

**Location**

String name
double latitude
double longitude
double radius

boolean isInLocation(double lat, double lon)

**Period**

long startTime
long endTime

**HighestTemperatureFact**

double temperature

# Design Task: **Service Concept**

heig-vd
Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud

***Who*** *are your users?*

***What*** *problem are you solving for them?*

***How*** *do you plan on solving it?*

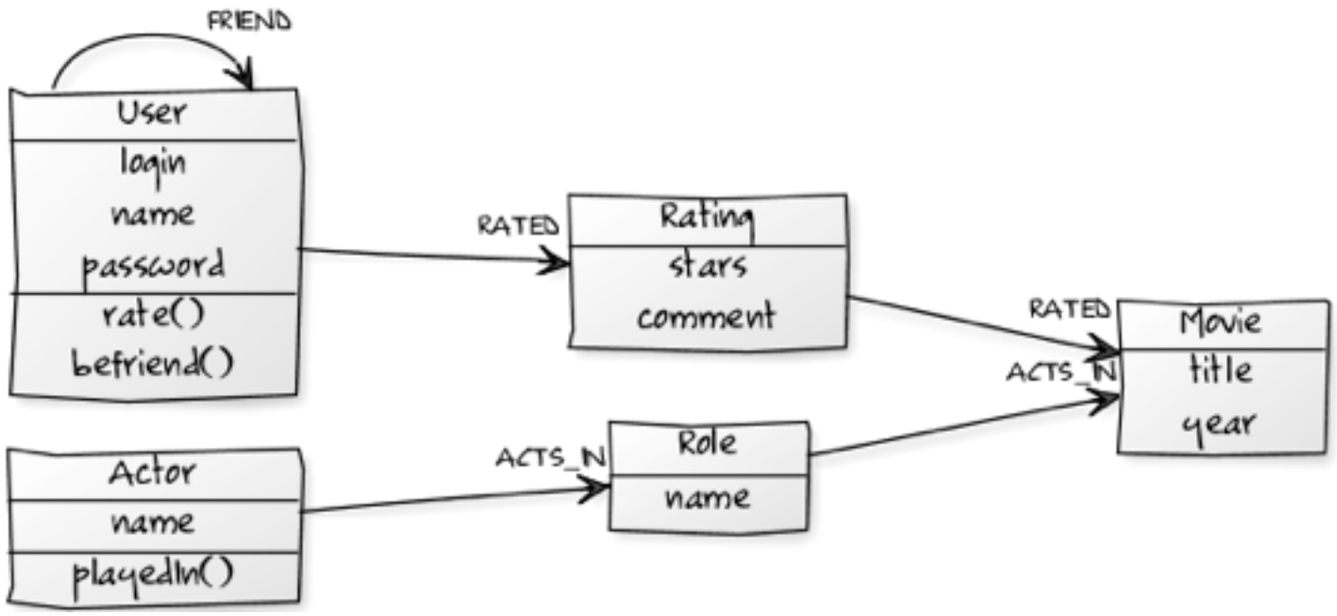
***How*** *will users interact with your service (sensors, objects, app, etc.)?*


http://www.brunowong.me/portfolio/music-dojo-web-page-mobile-app/

# Design Task: **Domain Model & Entities**

heig-vd
Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud

- What is your **application domain**?
- What are the **entities** in your application domains domain?
- What are their **properties** (attributes)?
- What are the **relationships** (composition, aggregation, inheritance) between these entities?




http://docs.spring.io/spring-data/data-neo4j/docs/3.0.1.RELEASE/reference/html/tutorial_domain.html

# Design Task: **Sensors**

heig-vd
Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud

- What **types of sensors** do you plan to use for your service?
- Are they measuring something in the **physical** or in the **digital** world?
- **What** do they measure or detect?
- At which **frequency** do they take and report observations?
- What will you do if you want to **simulate** or implement these sensors?

**sensor** | ˈsɛnsə |

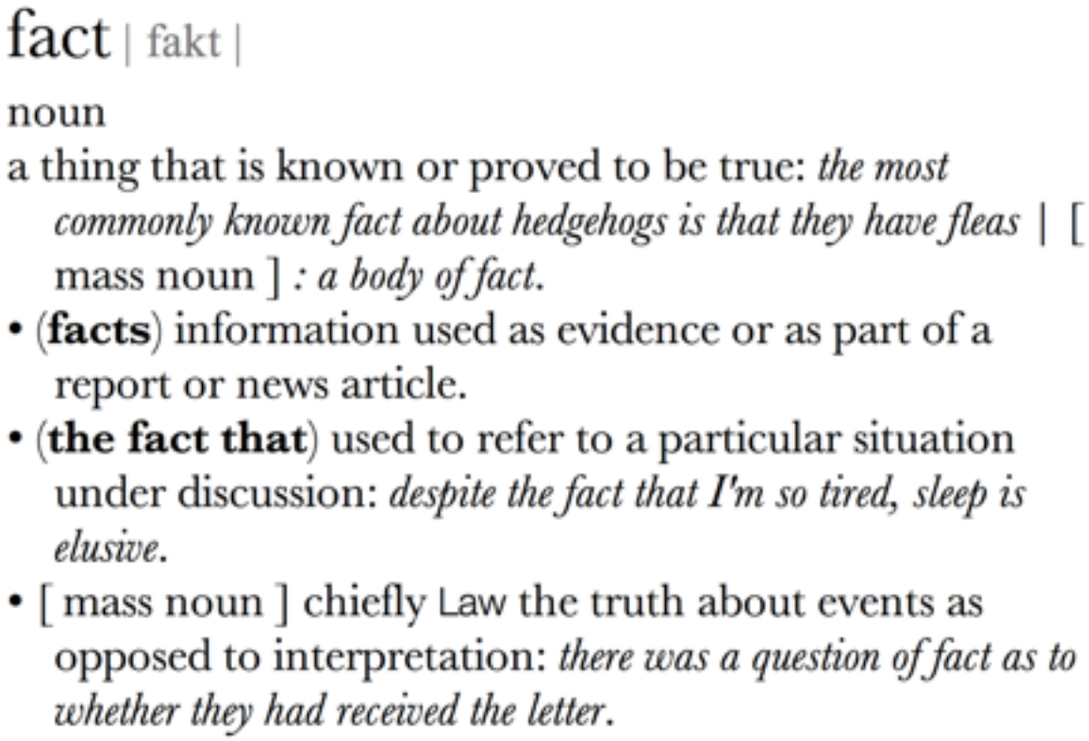
noun
a device which detects or measures a physical property and records, indicates, or otherwise responds to it.

ORIGIN 1950s: from SENSORY, on the pattern of *motor* .

# Design Task: **Facts**

heig-vd
Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud

- What **types of facts** about your entities do you need to capture, in order to deliver features to your users?
- For every type of fact, what are the **attributes** and what are the **relationships with entities**? (e.g. a `LastKnowPositionFact` might link a `Person` entity and a `Location` entity and have a timestamp attribute)



fact | fakt |

noun

a thing that is known or proved to be true: *the most commonly known fact about hedgehogs is that they have fleas* | [ mass noun ] *: a body of fact.*

- (**facts**) information used as evidence or as part of a report or news article.
- (**the fact that**) used to refer to a particular situation under discussion: *despite the fact that I'm so tired, sleep is elusive.*
- [ mass noun ] chiefly Law the truth about events as opposed to interpretation: *there was a question of fact as to whether they had received the letter.*