Naver Labs Europe and Domain-Specific Behaviour Definition

José Miguel Pérez **25th November 2019**

E-mail: jm.perez@naverlabs.com
Twitter: @jozemi

Outline

- About me
- Naver Labs Europe
- Domain-Specific Behaviour Definition

Outline

- **About me**
- Naver Labs Europe
- Domain-Specific Behaviour Definition

José Miguel Pérez Álvarez

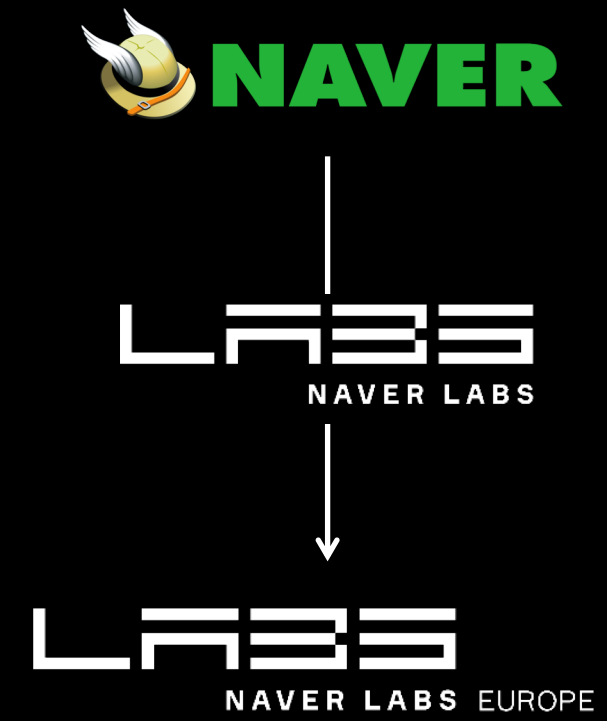
- Computer Engineering & Master Software Engineering (University of Seville) **MDE**
- Founder of Intelliment Security **MDE**
  - MBA
- PhD (Idea Research Group. University of Seville)
  - Title: Decision-Making Support for the Alignment of Business-Process-Driven Organizations with Strategic Plans **MDE**
- Research Scientific at Naver Labs Europe **MDE**

LABS

Outline

- About me
- **Naver Labs Europe**
- Domain-Specific Behaviour Definition

Naver corp



LABS

Naver

- Korean internet company
- Internet content service company (130 web and mobile services)
- 9th most innovative company in Forbes ranking (June 2018)
- Operates Korea's top search engine NAVER (75% market share)
- Frequently referred to as « the Google of South Korea » (Ref. Wikipedia)

[...]

LABS

Naver Labs

- Ambient intelligence (AmI) company innovating in contextual, location based and mobility services

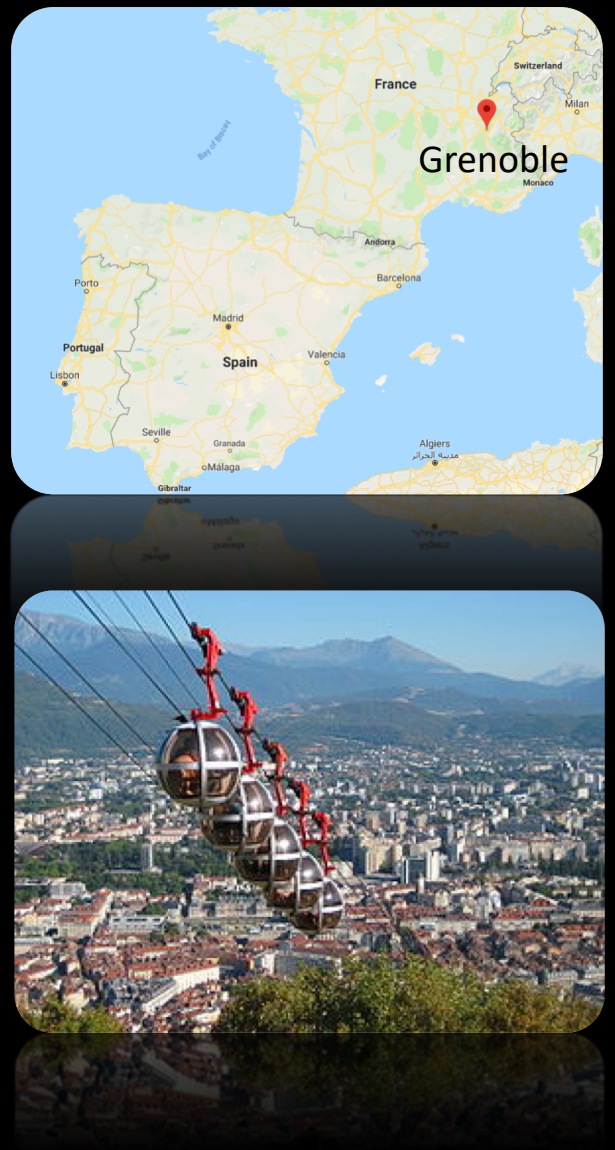
Autonomous driving / environment comprehension

5G Brainless Robot

Self-updating maps

Naver Labs Europe

- Located in Grenoble (France)
- The biggest industrial research centre in AI in France



Naver Labs Europe

- **Previously:** Xerox Research Center Europe
- Acquired by Naver in 2017

Naver Labs Europe

- Research Lines
  - Computer Vision
  - 3D Vision
  - Search and Recommendations
  - Machine Learning and Optimization
  - Natural Language Processing
  - UX and Ethnography
  - **Systemic AI**

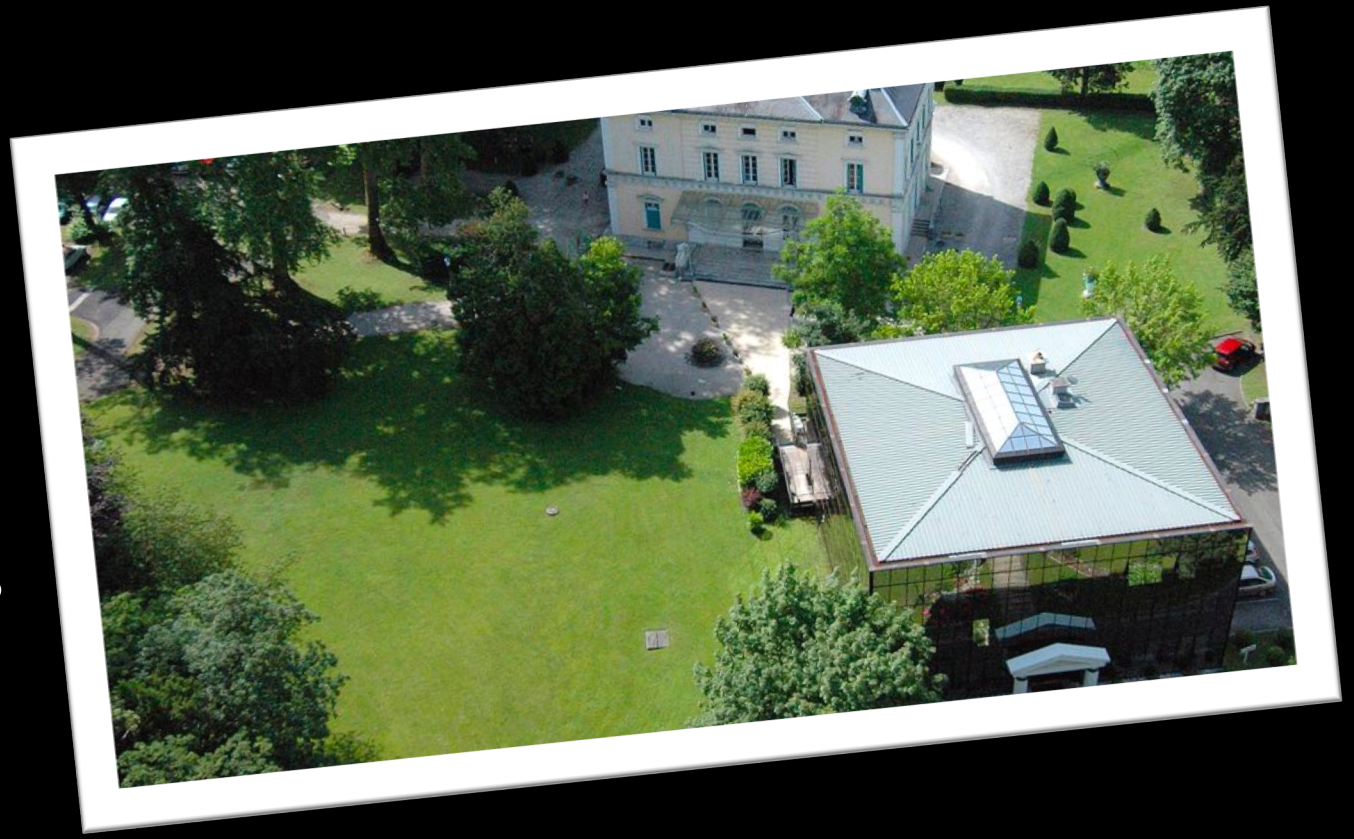
Naver Labs Europe

- **Systemic AI**
  - Software Engineering and/for AI
  - Large-scale systems
  - Data
  - Heterogenous systems
  - AI/ML Components
  - Integration
  - Reusability
  - Modelling
  - Behaviour

Naver Labs Europe

- **Opportunities**
  - PhD Students
  - Internships programs
  - Research positions

- We are open to collaborations

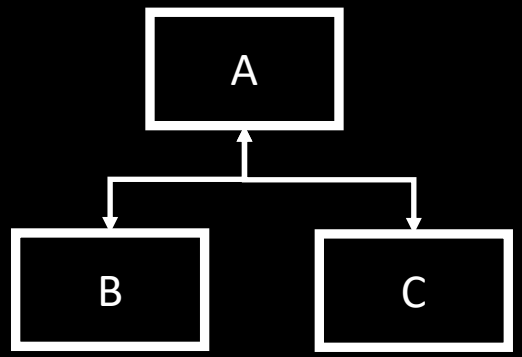


Outline

- About me
- Naver Labs Europe
- **Domain-Specific Behaviour Definition**

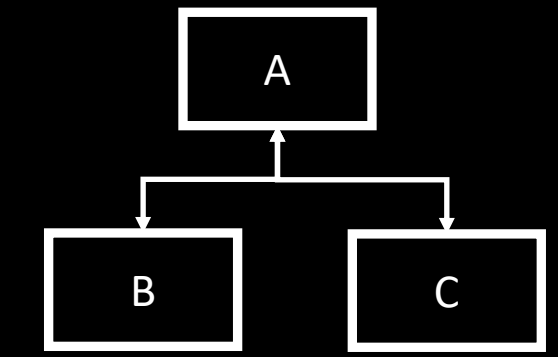
Context and motivation

- From: Application design
- To: Running systems



Context and motivation

- Task just for technical people with knowledge
- Time consuming
- Error prose
- Involve
  - Programming
  - Configuration
  - Building
  - etc. etc. etc.



Application design

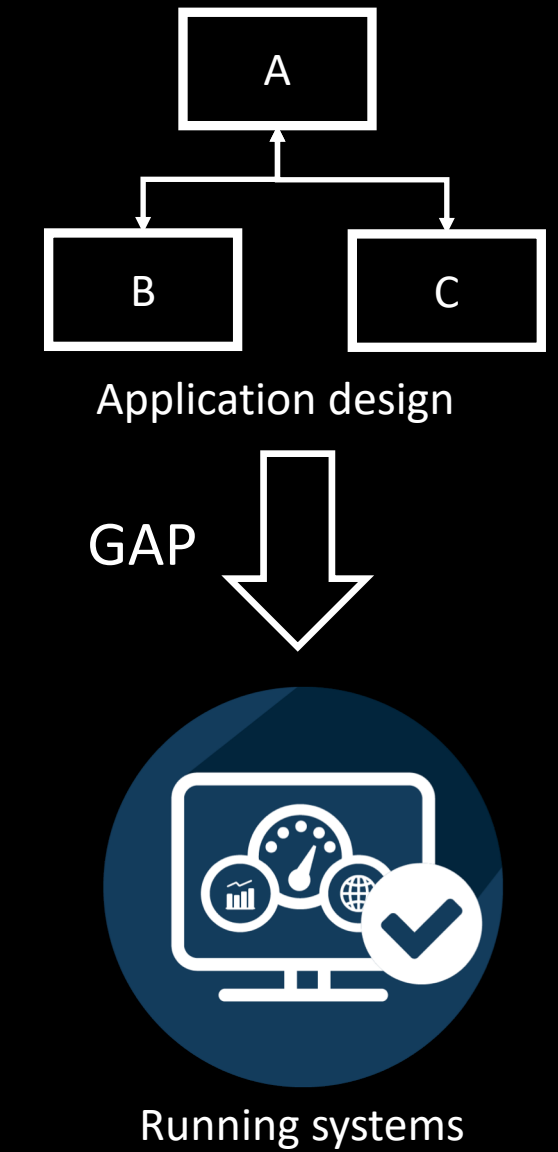
GAP

Running systems

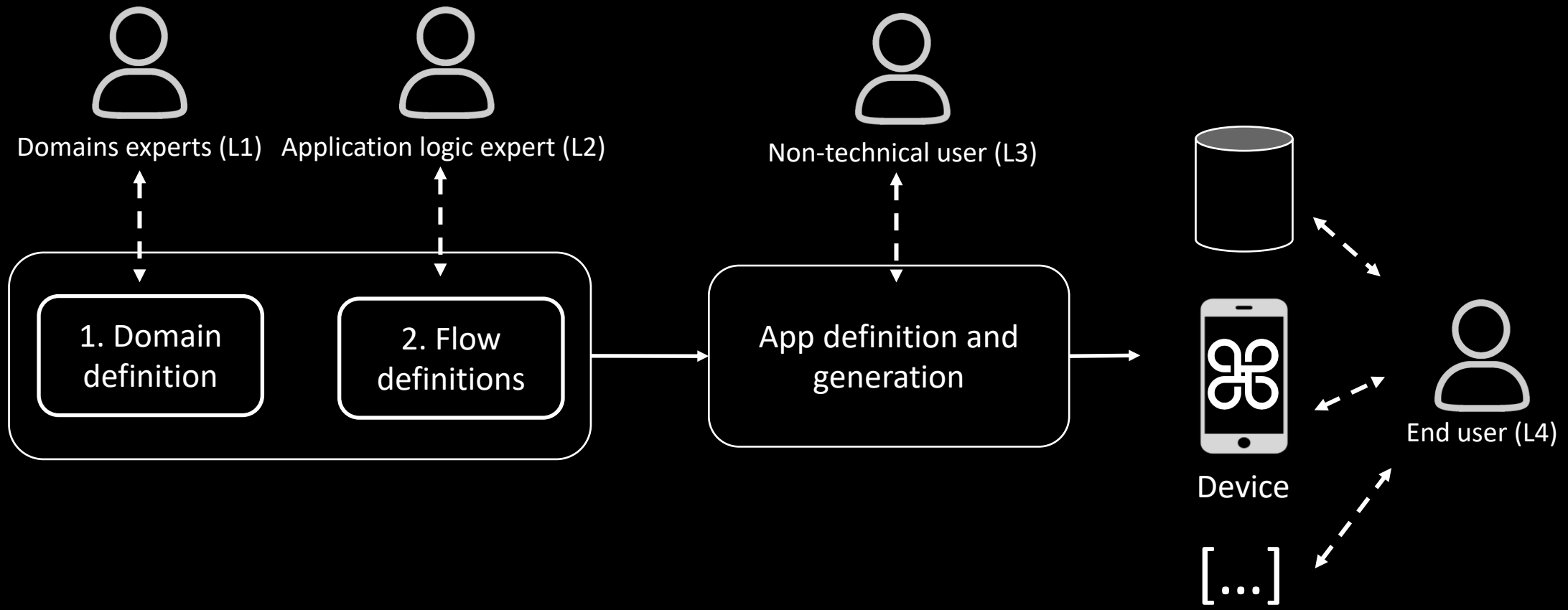
Context and motivation

- We are working on: Reducing that GAP
  - Less complex task
  - Let is be accessible for non-technical users
  - Less error prose

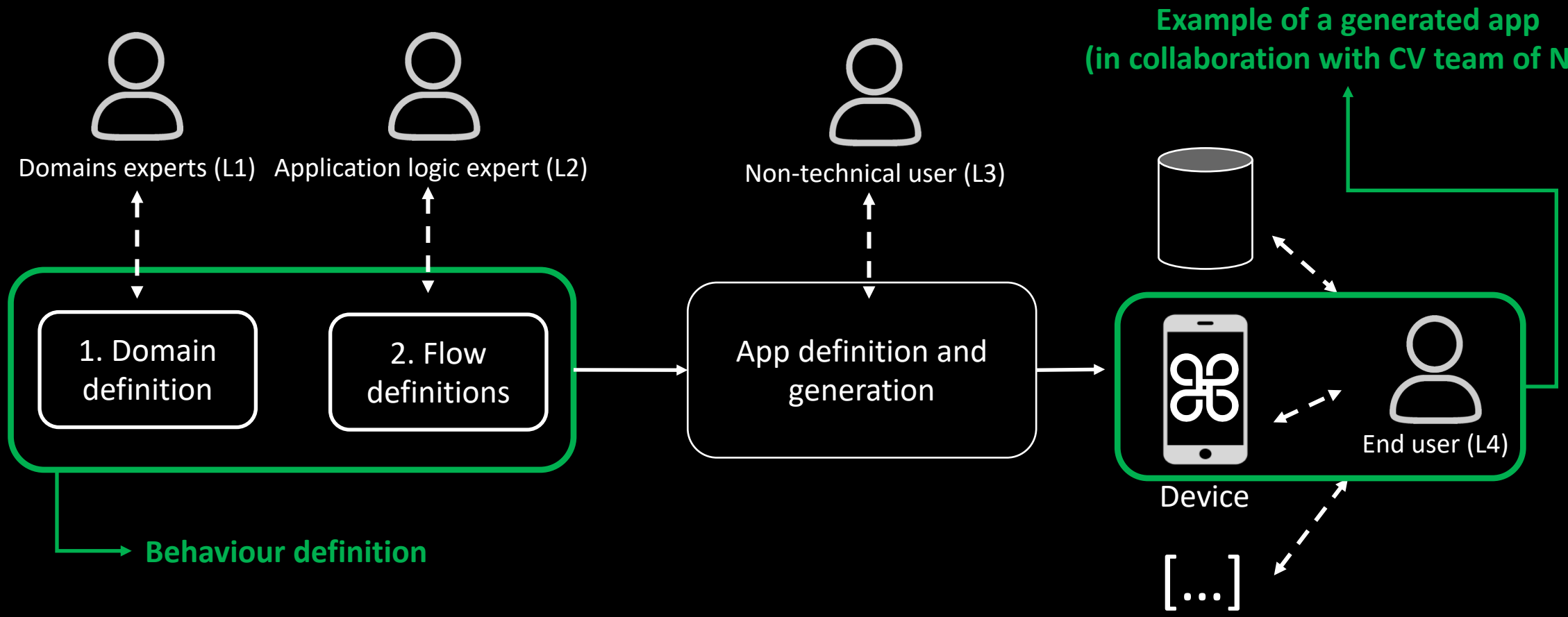
- Currently: Automatic app generation



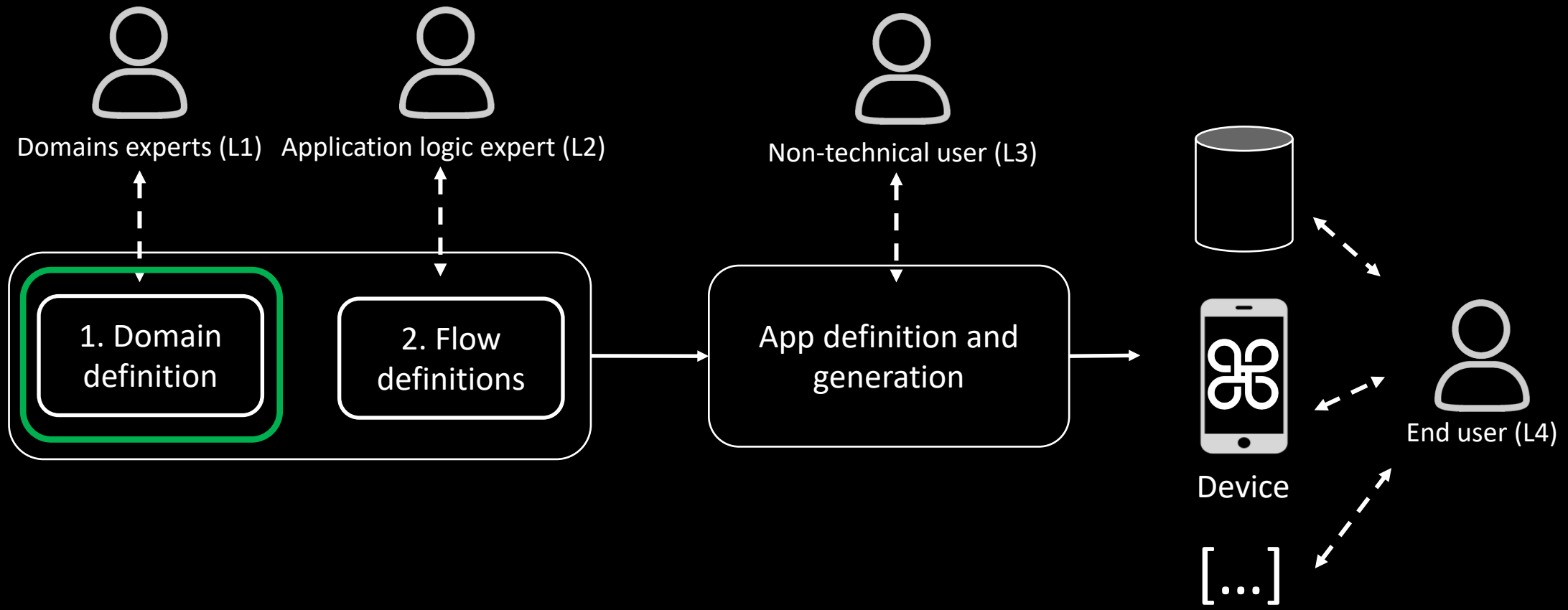
Approach



Approach



Approach



Domain

- Set of basic units of behaviour (activities)
- Ready to be executed
  - Inputs / Outputs
  - Data descriptions
  - External services
  - User interactions
- Of a determinate domain
- Requires good understanding of technology

Domain meta-model (Emfatic style)

```
@namespace(uri="http://naverlabs.com/flow/models/domain", prefix="domain")
package domain;

class DomainDefinition  {
  val DSActivityType[*] activityTypes;
  val DSService[*] services;
  val IO[*] ios;
  val Type[*] types;
  val DSServiceRelation[*] dsServiceRelations;
  val IORelation[*] ioRelations;
}
```

```
class IO extends GovernedObject{
  val Field[*] fields;
}

class Type extends GovernedObject {
  val Attribute[*] attributes;
}
```

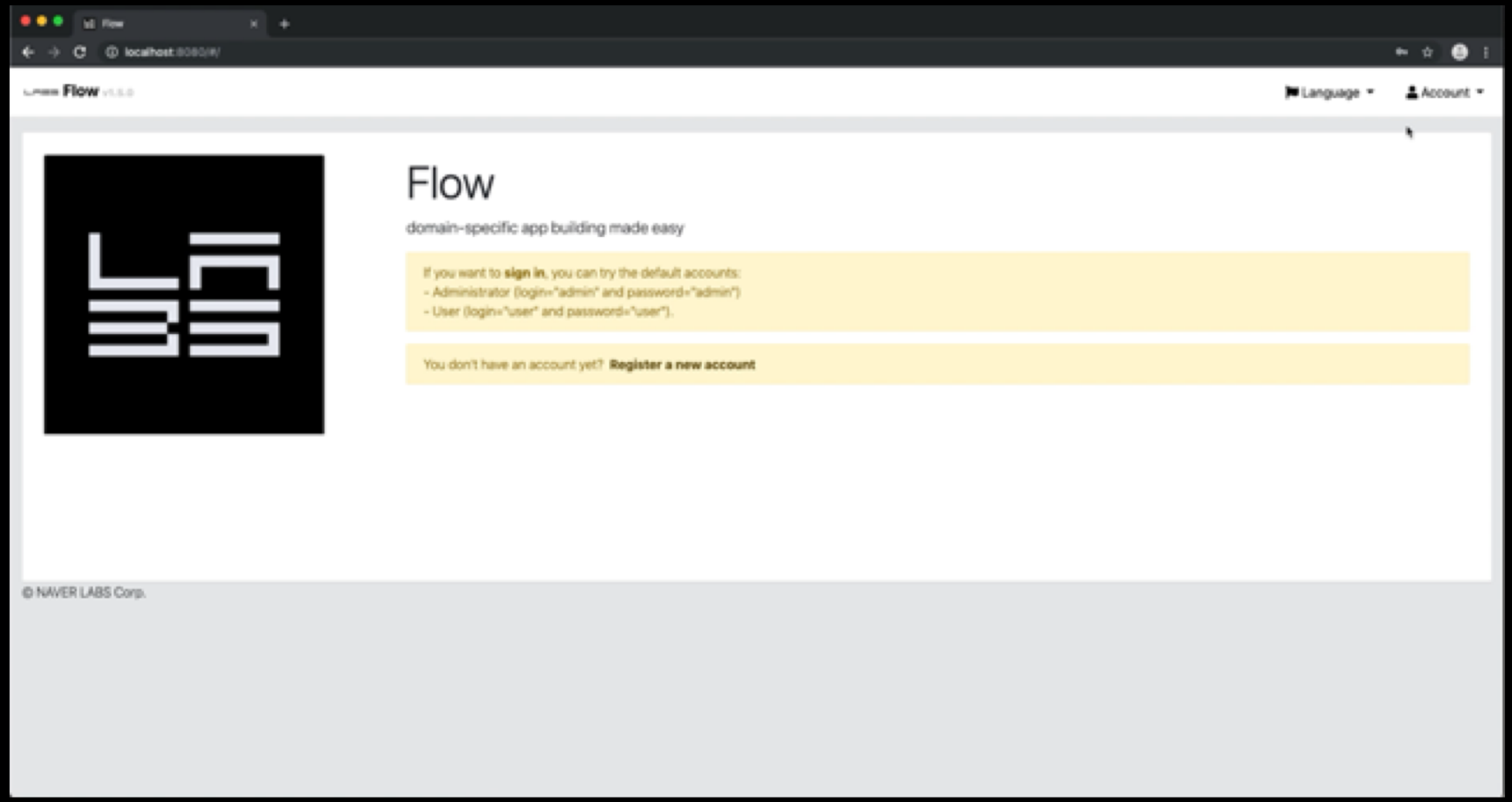
```
class DSService extends GovernedObject {
  val Parameter[*] inputs;
  val Parameter[*] outputs;
}
```

```
class DSServiceRelation extends DSActivityTypeRelation {
  ref DSService[1] dsService;
  val InputsServiceMapping[*] inputMappings;
  val OutputServiceMapping[*] outputMappings;
}
```

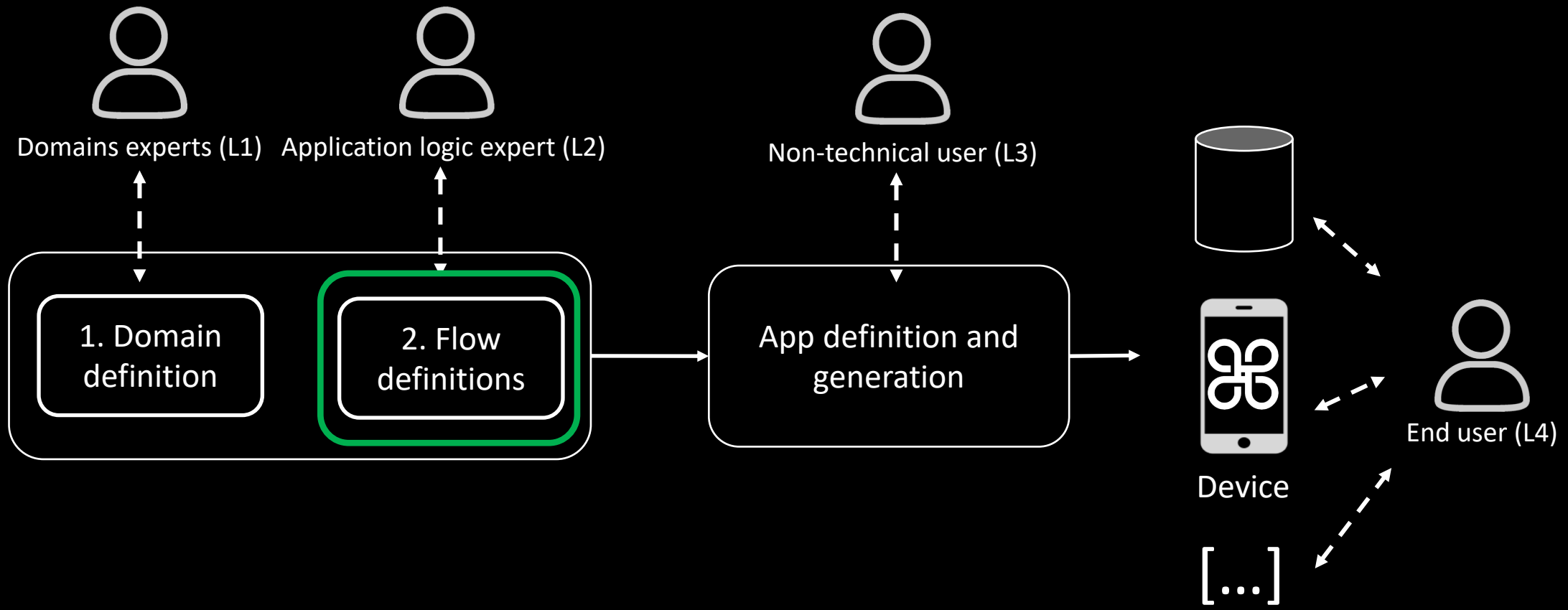
Examples of Domains and activities

- Tourist office domain
  - Obtain location of a tourist
  - Get the closest POI
  - etc.
- Payment domain
  - Check credit card
  - Withhold money
  - Make a payment
  - etc.
- Domain to process image
  - Create a thumbnail
  - Put a bounding box
  - etc.
- CV domain
  - Detect number of people in image
  - Look for similar images
  - etc.

Modelling domains



Approach



Flow

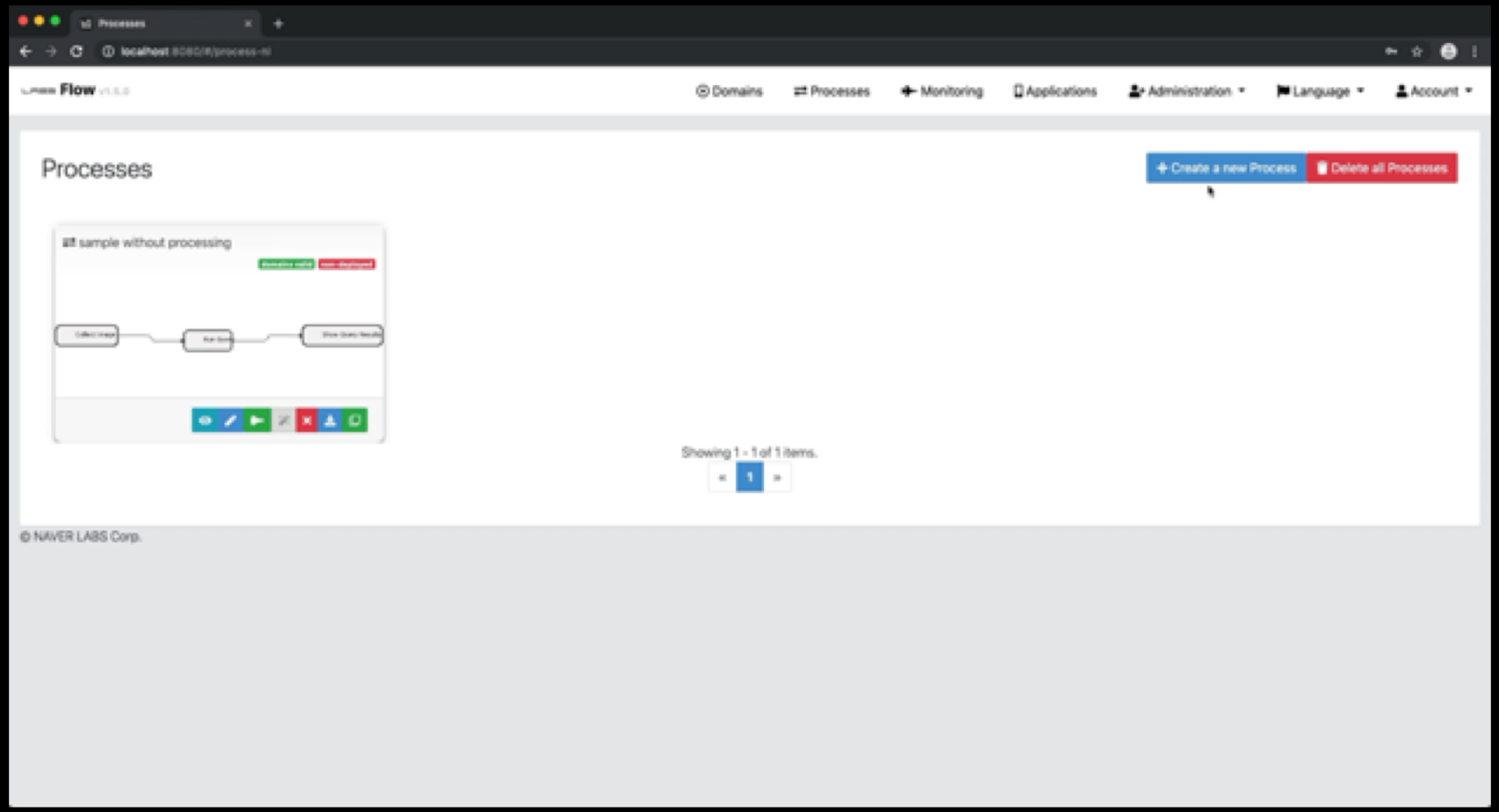
- Models a complex behaviour
- Relates activities available from one or many domains
  - Direct
  - Conditional
- Less technical task than defining a domain

Flow meta-model (Emfatic style)

```
@namespace(uri="http://naverlabs.com/flow/models/mangrovito", prefix="mangrovito")
package mangrovito;

import "platform:/resource/com.naverlabs.flow.model.domain/src/main/resources/domain.ecore";

abstract class Element {
  attr String ~id;
}

class Flow extends Element{
  attr String name;
  val Step[*] steps;
  val Transition[*] transitions;
}

abstract class Step extends Element{
   attr String name;
}

class StartStep extends Step{

}

class ActivityStep extends Step {
  ref domain.DSActivityType dSActivityType;
  val ForcedValue[*] forcedValues;
}
```


LABS

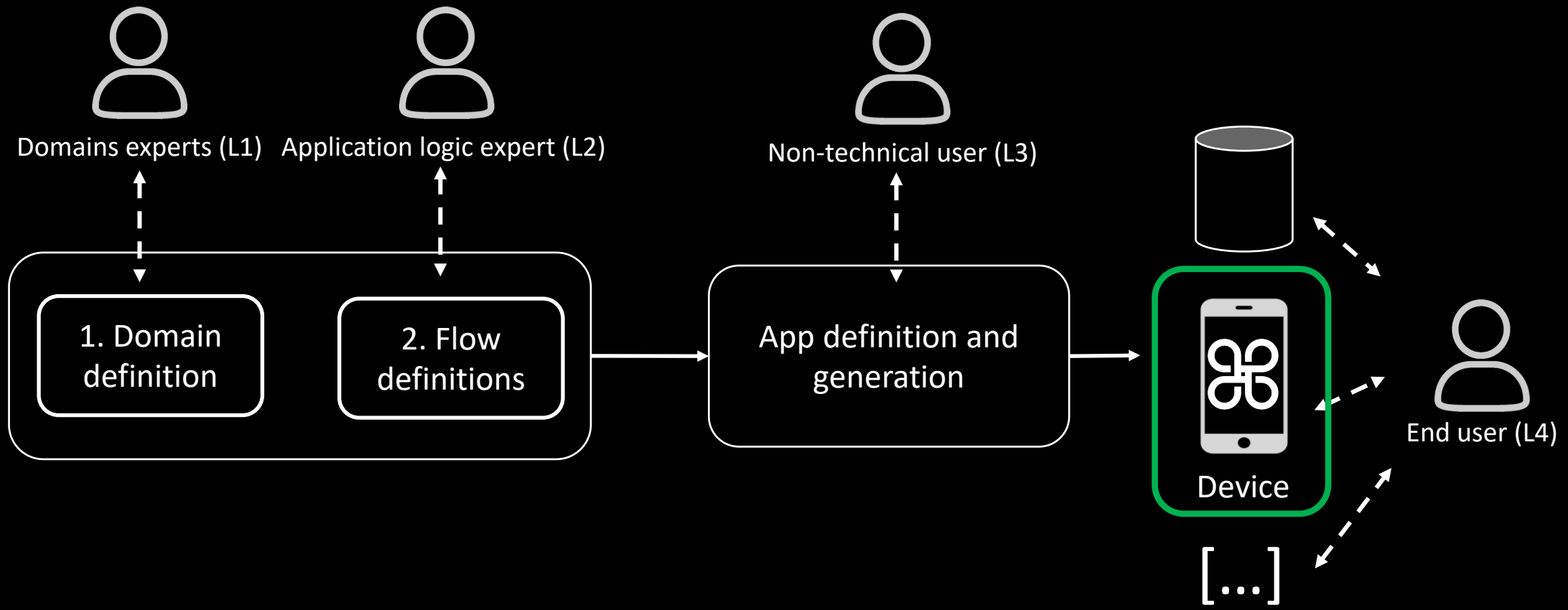
Modelling flows



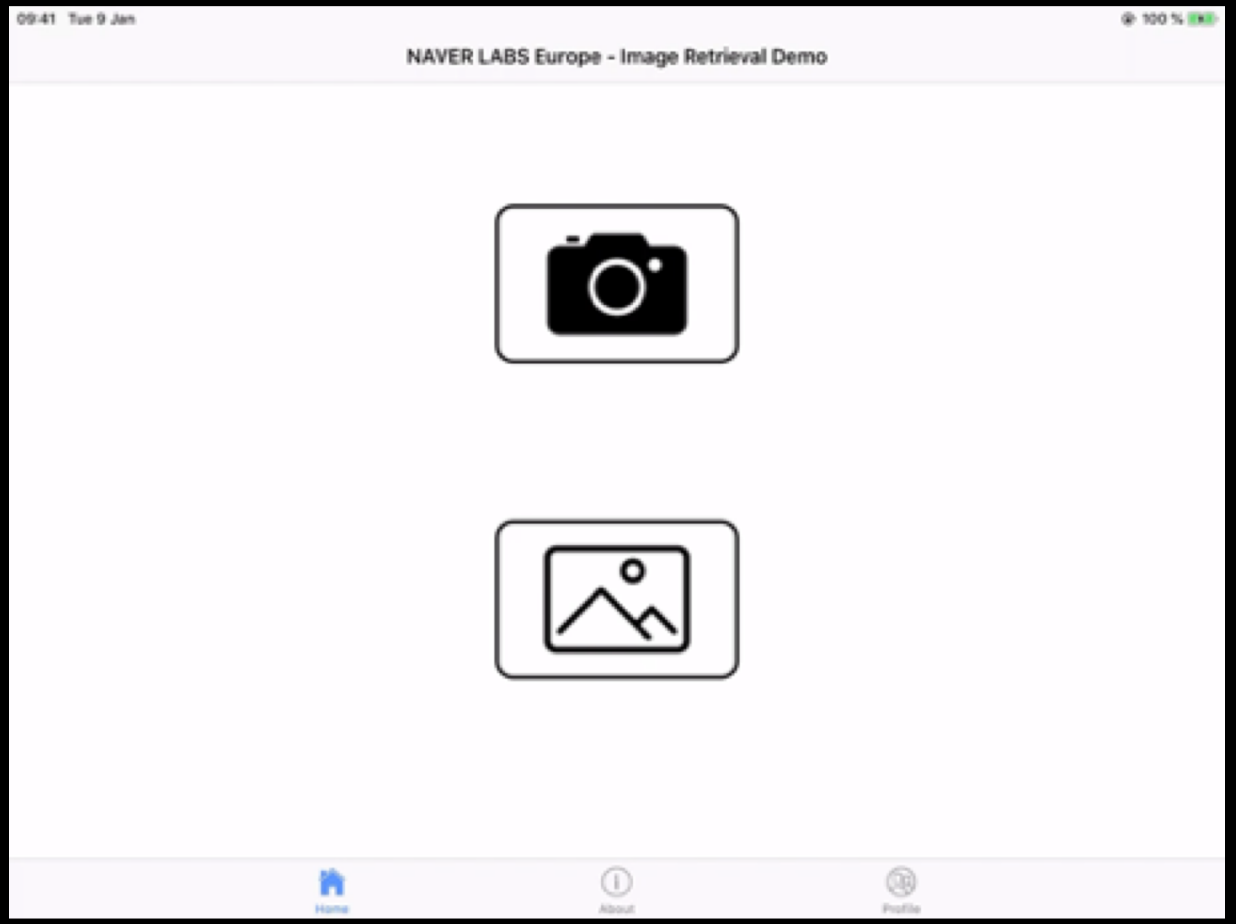
Ok, now we have defined behaviour

- Previously:
  - To implements domain-specific editors for BPM models
  - Transform this behaviour to BPM
  - Execute by using a BPM engine
- Currently
  - We have our own engine
  - To define the behaviour for end-user apps

Approach



Generated application



Thank you!
Questions?

José Miguel Pérez **25th November 2019**

E-mail: jm.perez@naverlabs.com
Twitter: @jozemi