IOT METHODOLOGY 1-2-3

Dr. Frank Puhlmann
frapu.de/iot

Table of contents

0. Enterprise IoT Basics & Getting Started

1. **Plan** a Project
   - ▶ Stakeholders & Roles
   - ▶ Domain Model
   - ▶ Processes & Rules
   - ▶ User Interfaces
   - ▶ Connectivity
   - ▶ Finalize everything together
     - Project Sketch
     - Architecture Blueprint
     - Wrap Up
2. **Build** a Project
3. **Run** a Project

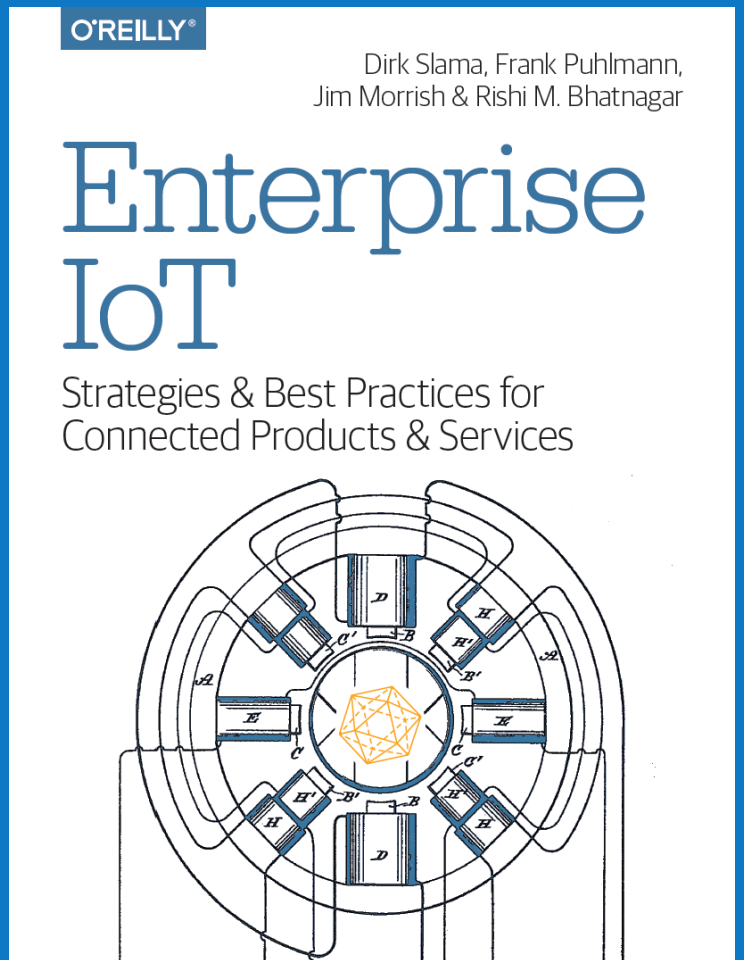
Structure

After the initial draft, refinements of your plan come into play that will be discussed in the planning sections.

Once you have your planning refined so that you can start building something (thing of Agile!), you can move to the build phase and iterate.

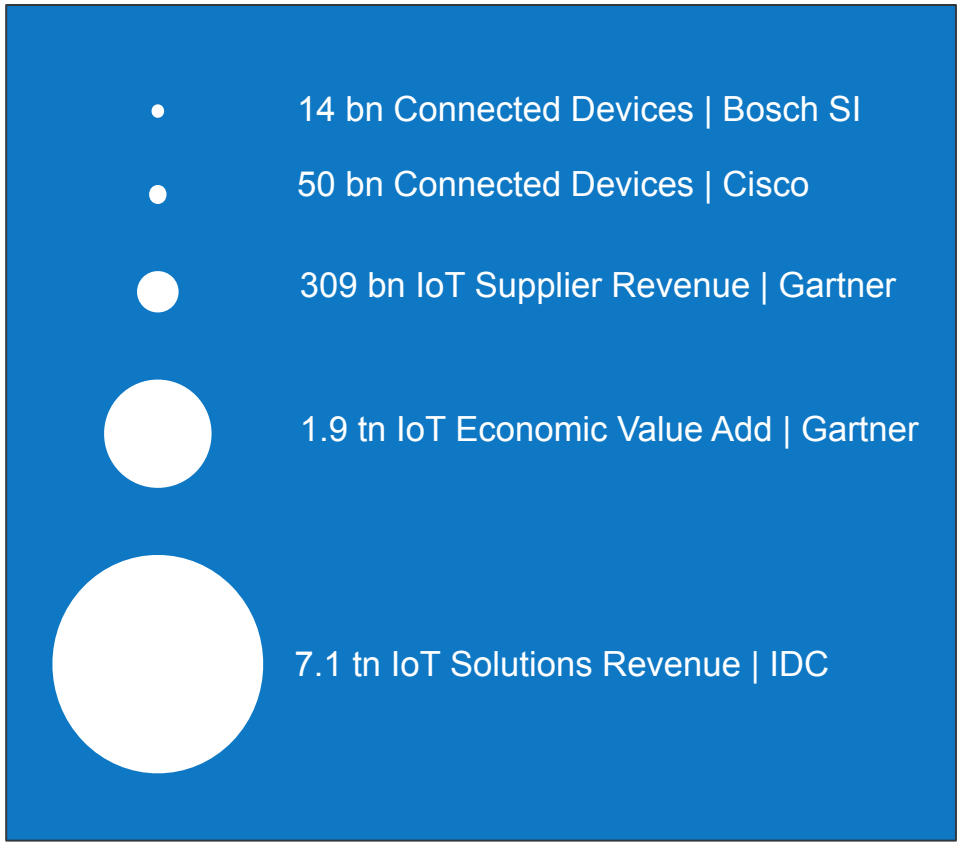
Build, and Run, however, are out of scope for this deck.

ENTERPRISE IOT BASICS



IOT PREDICTIONS

Some Big Numbers:



Data from http://postscapes.com/internet-of-things-market-size

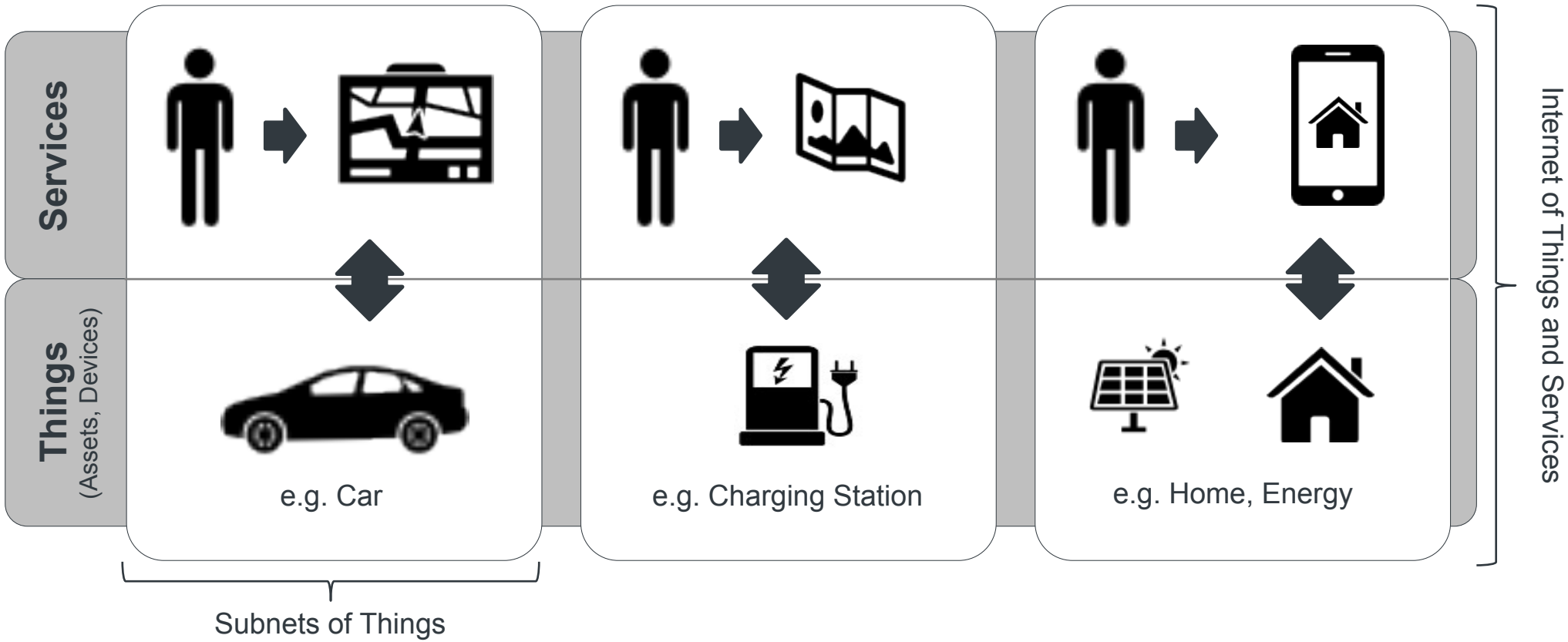
Some Small Numbers:

Peter Middleton, Gartner:
By 2020, component costs will have come down to the point that connectivity will become a standard feature, even for processors costing less than

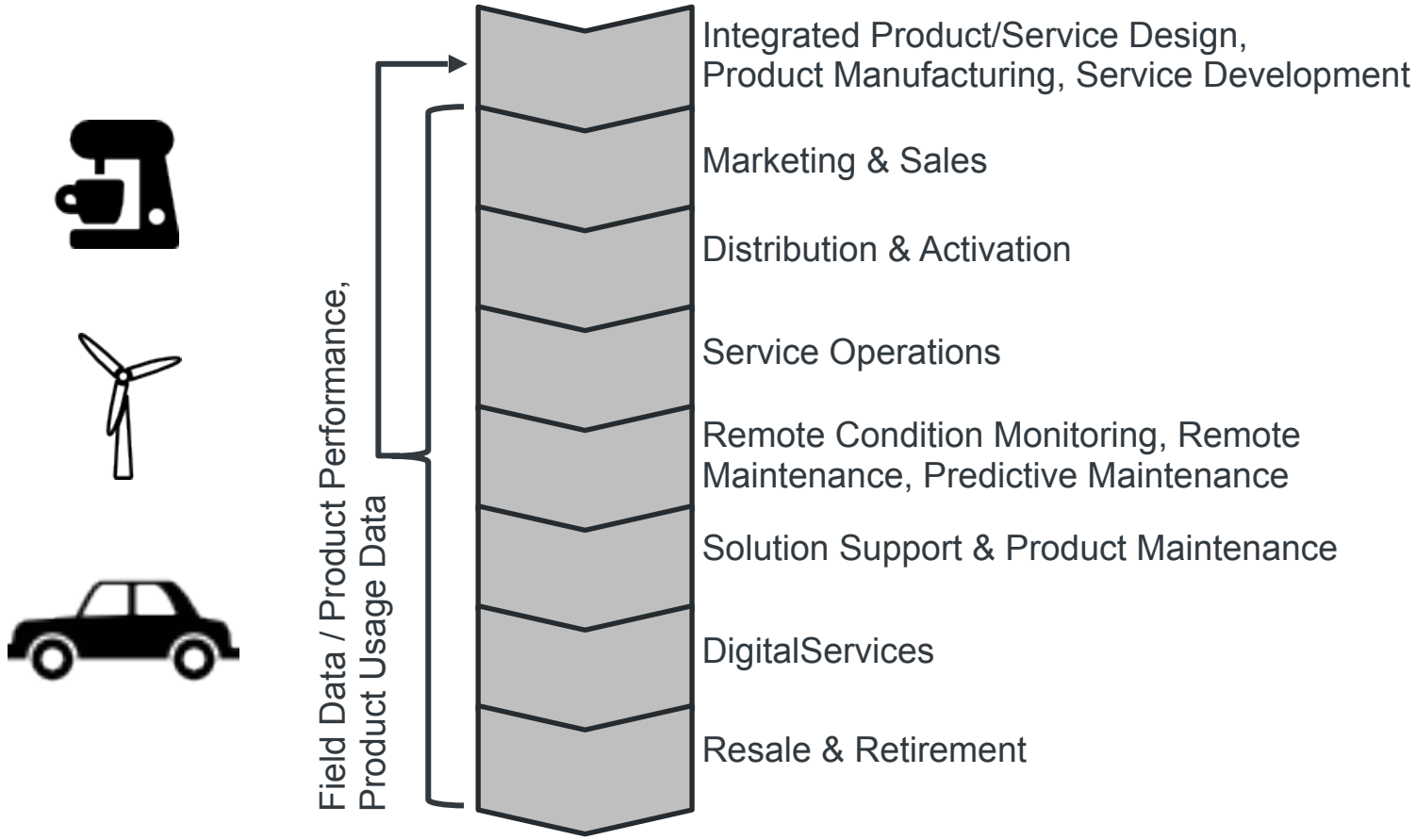
$1

Source: www.enterprise-iot.org

INTERNET OF THINGS ANS SERVICES

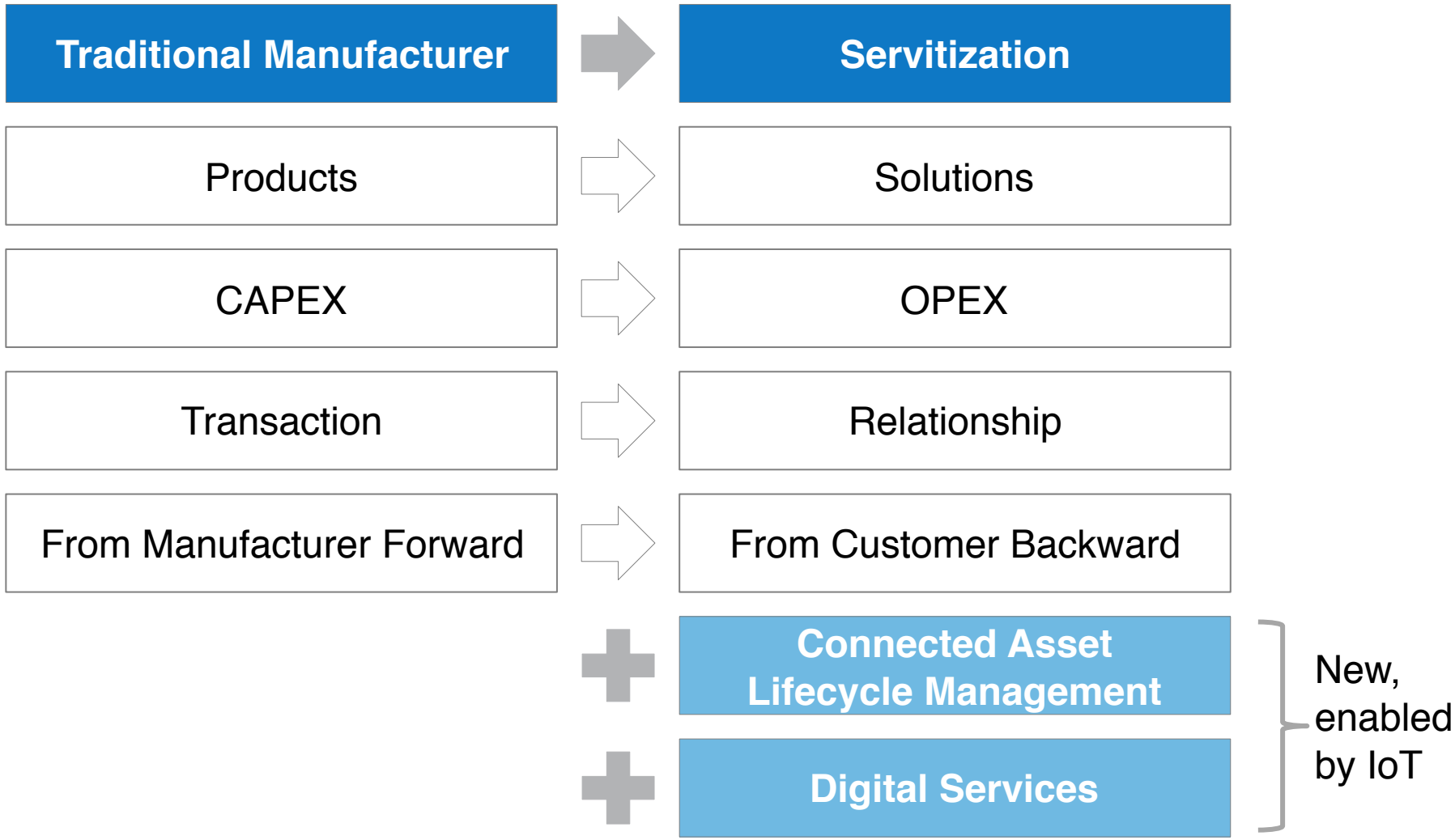


Source: www.enterprise-iot.org



Source: www.enterprise-iot.org

SERVITIZATION



Source: www.enterprise-iot.org

CLASH OF TWO WORLDS

“Machine Camp”

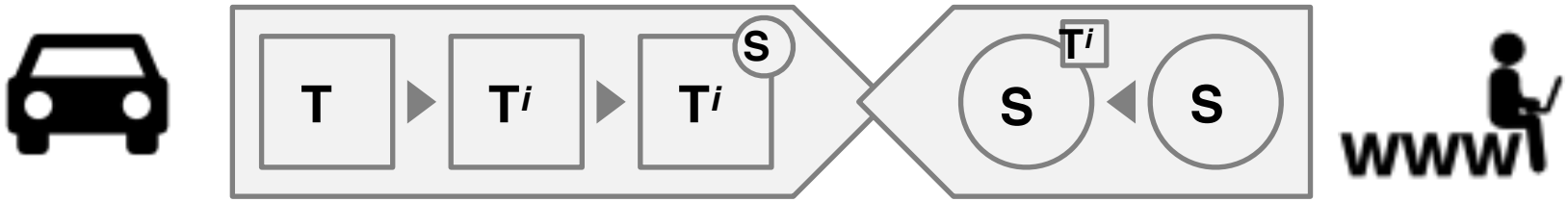
- “Brown field”
- Strong company heritage, risk aversion
- Corporate career is the norm
- Domains: Physics, engineering
- “Think big”
- Waterfall approach
- Standards like DIN/ISO
- Long QA & release cycles (“defect free”)
- Long lead times

“Internet Camp”

- “Green field”
- High-risk, VC-driven culture
- Entrepreneurial management and employees
- Domains: IT, services
- Focus on point solutions/MVP
- Agile approach
- Open source
- Perpetual beta (“Fast patches”)

Source: www.enterprise-iot.org

BIG PICTURE



"Machine Camp" | "Things" | Intelligent "Things" | Intelligent "Things" with Internet Service | Internet Service with (Intelligent) "Things" | Internet Service | "Internet Camp"

"*First, we equipped our light bulbs with sensors, so they would only be on if somebody was in the room. Next, we added an iPhone app so you can manage all of the electric lights in your home.*"

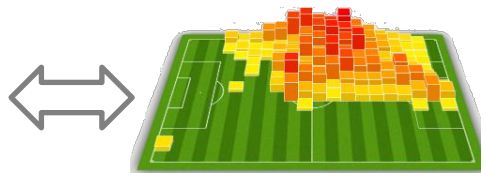
"*Our online security service is now using connected light bulbs to simulate a realistic lighting pattern when you are away from home at night.*"

Source: www.enterprise-iot.org

SOLUTION DESIGN

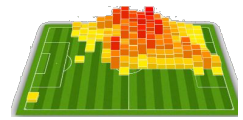
Real World
- Analysis
- Main “theories,” Business requirements

Digital Model
- Granularity required for business model
- Domain Model


Possible Inception Points
- Sensors, cameras, existing data streams

Reconstruction
- Creating the digital model

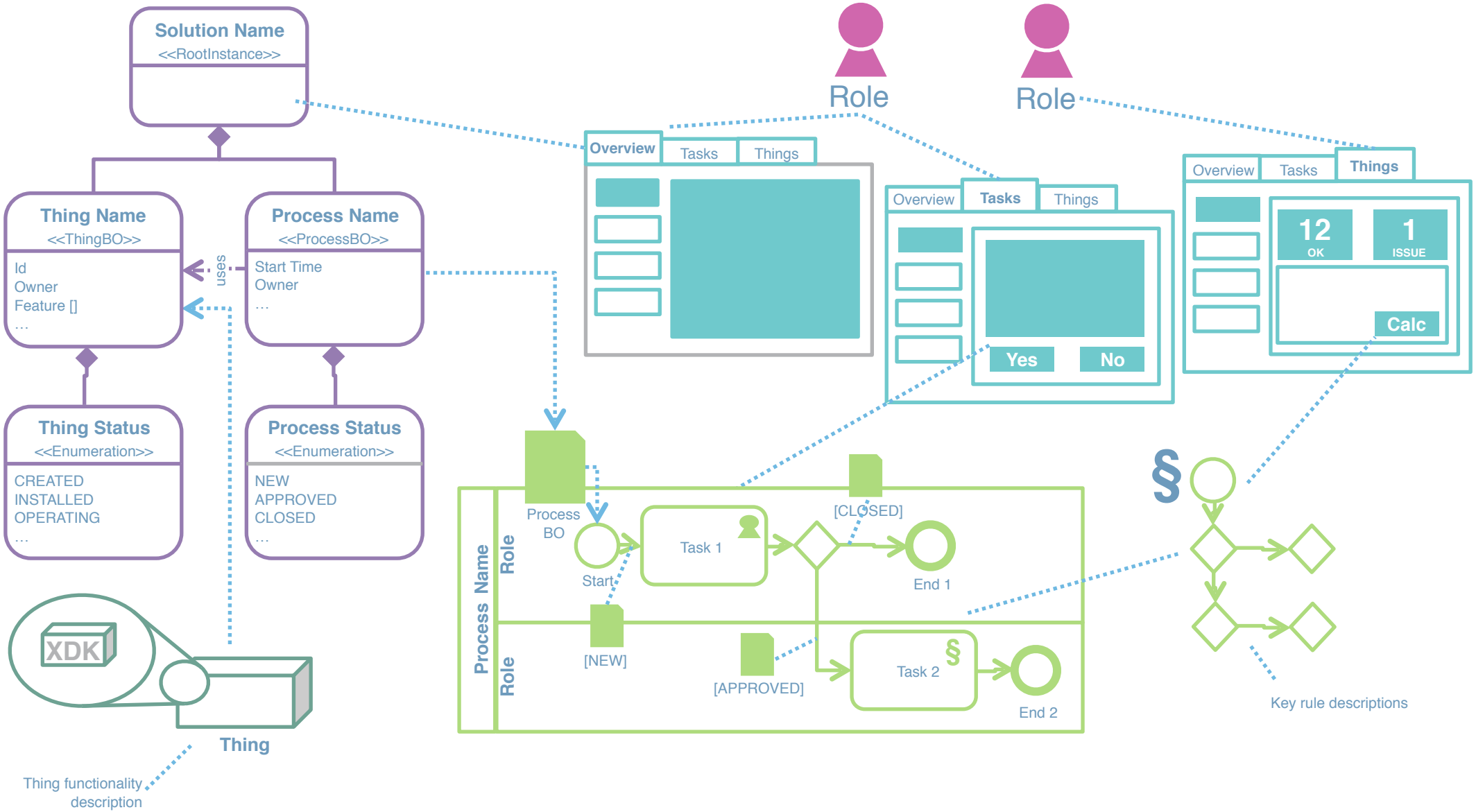



Solution Design
- “Sketch”: Business architecture, component architecture, asset integration architecture, technical infrastructure, hardware design

Source: www.enterprise-iot.org

GETTING STARTED

SOLUTION DRAFT



STAKEHOLDERS & ROLES

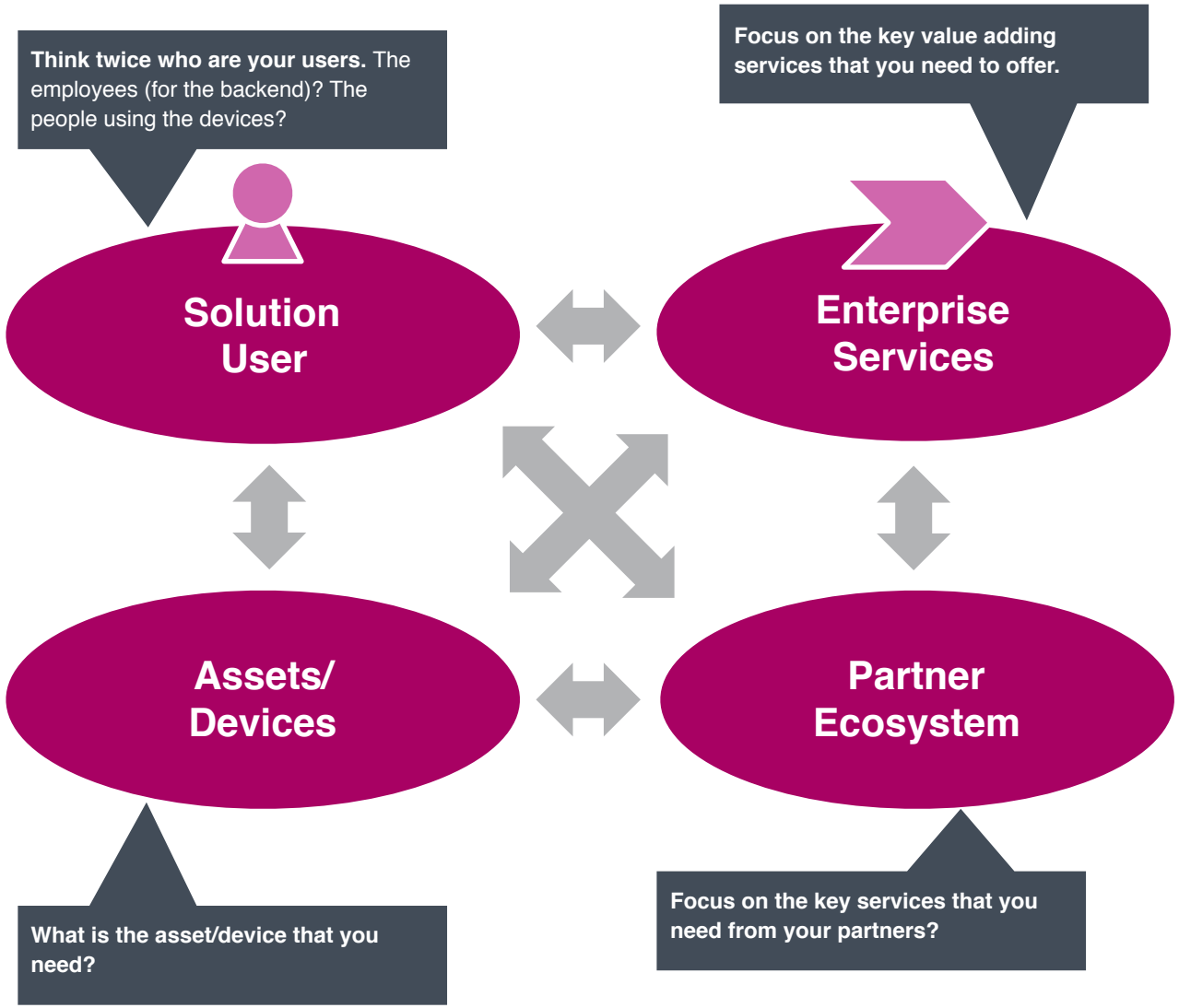
STAKEHOLDERS

How to capture the four most important stakeholders:

- **Solution Users:** The key persons using your solution
- **Enterprise Services:** The key services provided by the enterprise in charge of offering the solution
- **Partner Ecosystem:** The partners that you need to run your solution
- **Assets/Devices:** The key things that you need for your solution

Keep also track of the relationships between the key stakeholders!

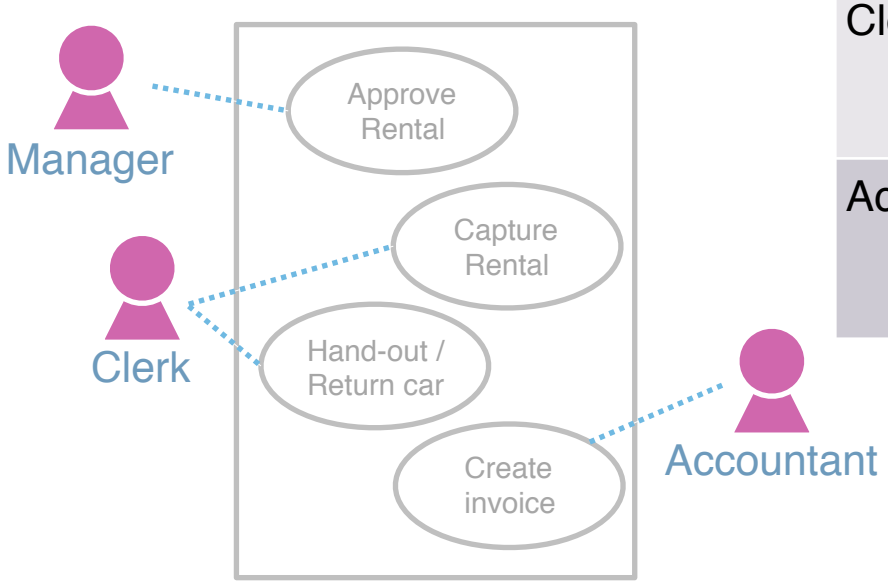
Don't forget the relationships!



ROLES

How to refine the roles:

- Capture the key aspects in a **table**
  - Name
  - Description
  - Number of users having this role initially (Field PoC)
  - Number of users for a production rollout
- Capture key use case in a **UML use case** diagram



Name	Description	Count Initially	Count Planned
Manager	Is responsible for the operations of the rental company office.	1	50
Clerk	Is responsible for customer care and rental support.	3	200
Accountant	Is responsible for creating the invoices.	1	10

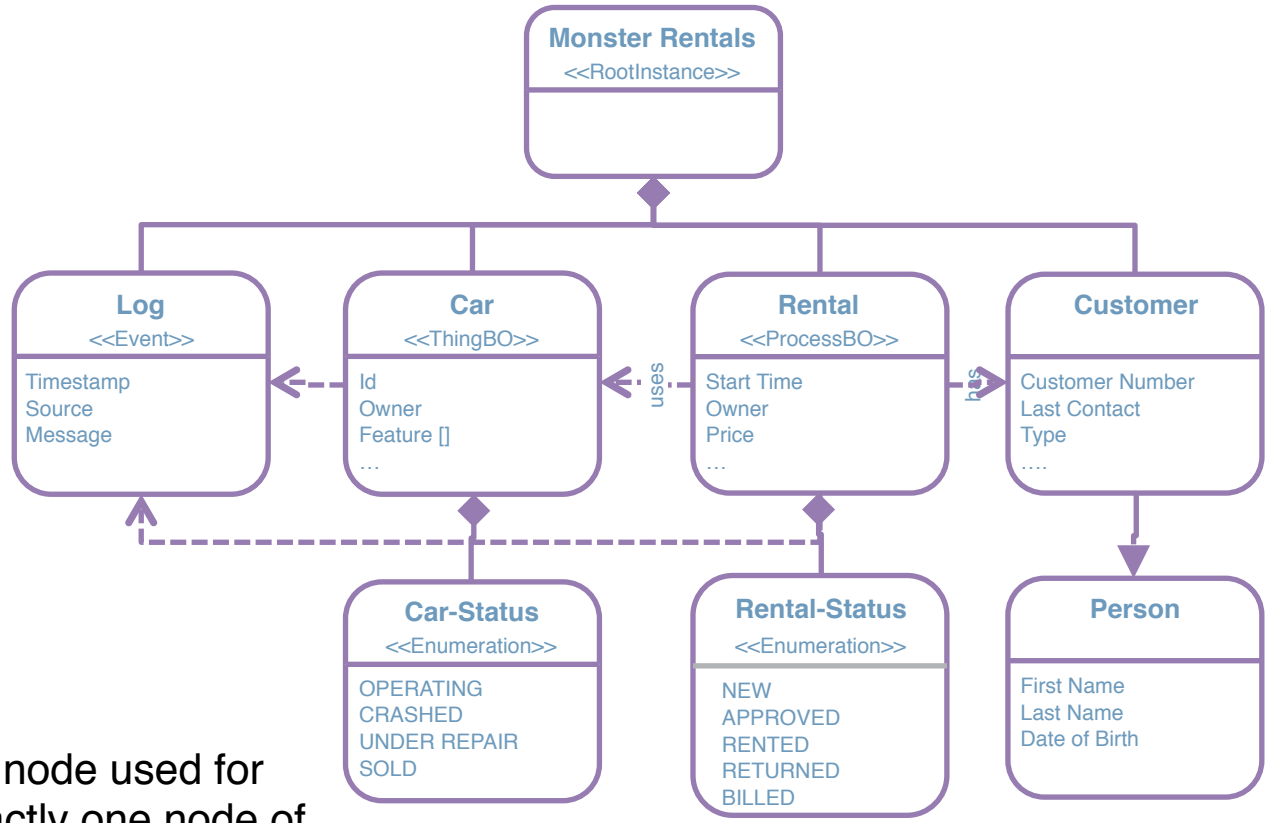
Where is the customer? We omitted the customer for the demo since he is not directly working with the system but interacting via the clerk. If, however, there is also a self-service UI for the customer (app, kiosk, website, etc) we need to include him as well!

DOMAIN MODEL

DOMAIN MODEL

How to capture the domain model:

- Start with the **RootInstance** naming your solution
- Aggregate the directly navigable entities (**business objects**) below
  - Include things via **ThingBO** and processes via **ProcessBO**
  - Feel free to add domain specific classes like „Customer“ etc.
- Create subclasses via further aggregation or inheritance
- Link other classes via associations



Key entities (Stereotypes):

- The **RootInstance** represents the root node used for navigation the BO. There is always exactly one node of this type.
- A **ProcessBO** represents a process business object.
- A **ThingBO** represents a (usually) physical thing or object.
- An **Event** represents something that happened with a thing or process (start, message, trigger, etc.)

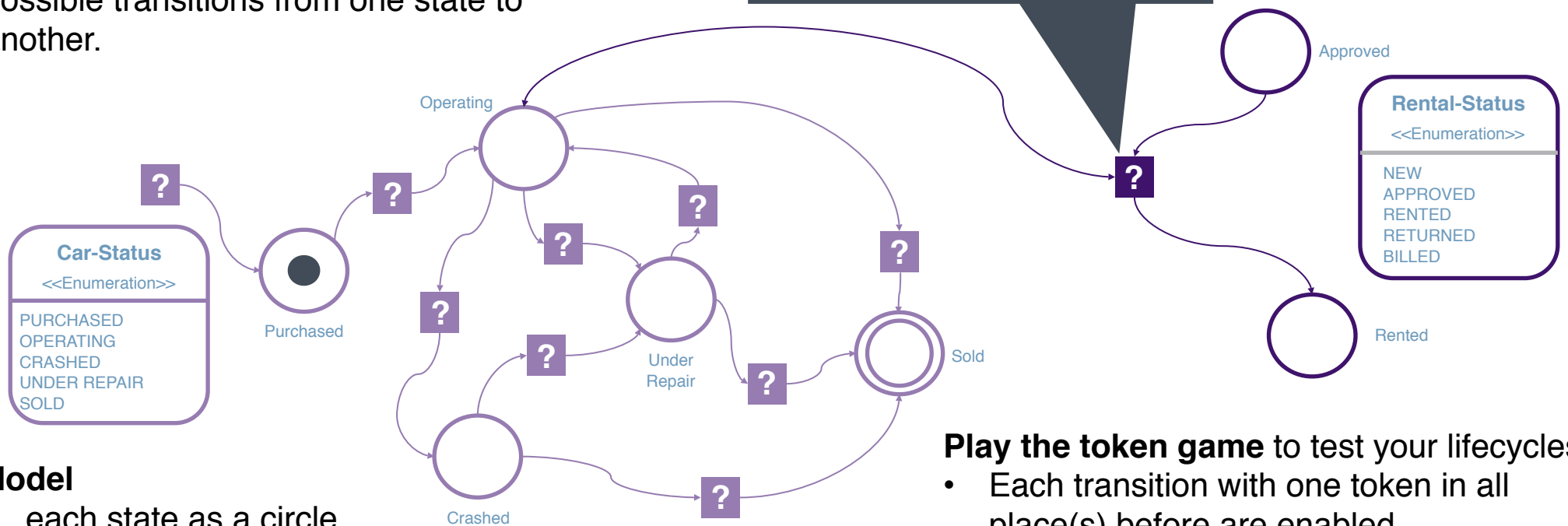
LIFE CYCLES

Keep track of the lifecycles of your entities. Make sure that you clearly understand the different states of your entities.

Having done that, discuss all possible transitions from one state to another.

Interlinked states between entities. If two states of an entity are interlinked, you can "combine" their lifecycles by "testing" via one entity is existing in specific state.

For instance, a rental can only transition from "Approved" to "Rented" if there is at least one operating car (simplified model!).



Model

- each state as a circle
- each possible transition as an arc and describe when/who/by whom it occurs

Play the token game to test your lifecycles

- Each transition with one token in all place(s) before are enabled
- You can remove these token(s) and put one into every outgoing place!

PROCESSES AND RULES

Business Process Diagrams (BPD) describe how state transitions of business objects must or should occur from a business point of view:

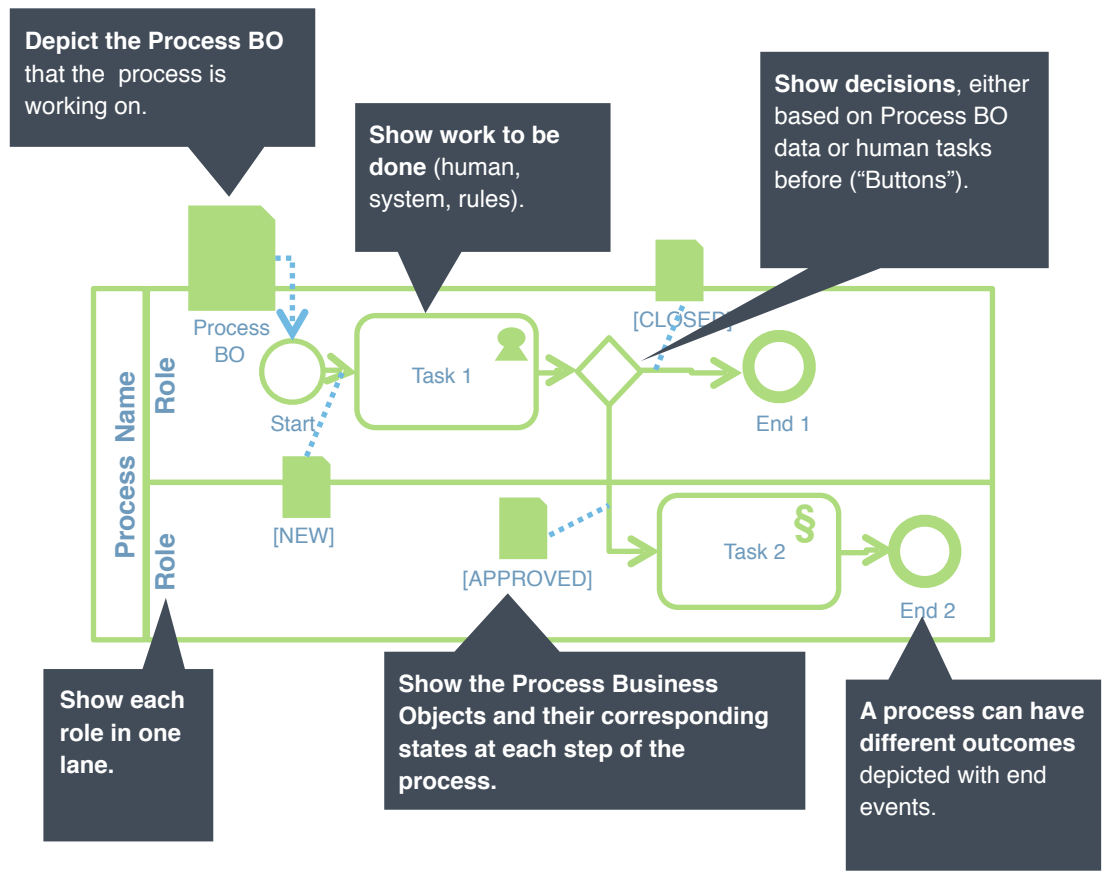
- Must: Processes are the only mean to achieve state transitions of business objects
- Should: Processes are a recommended way of how state transitions of business objects are handled

Different types of tasks

- Human Task: A task performed by a human via a frontend.
- Service Task: A task performed by another system.
- § Rule Task: A calculation done by a rule engine.

Business Processes are stateful and usually long running.

BPDs are completely standardized by the OMG's BPMN 2.0 specification.
See an overview online: http://bpmb.de/poster



RULES

Business Rules describe functional business logic and should be stateless.

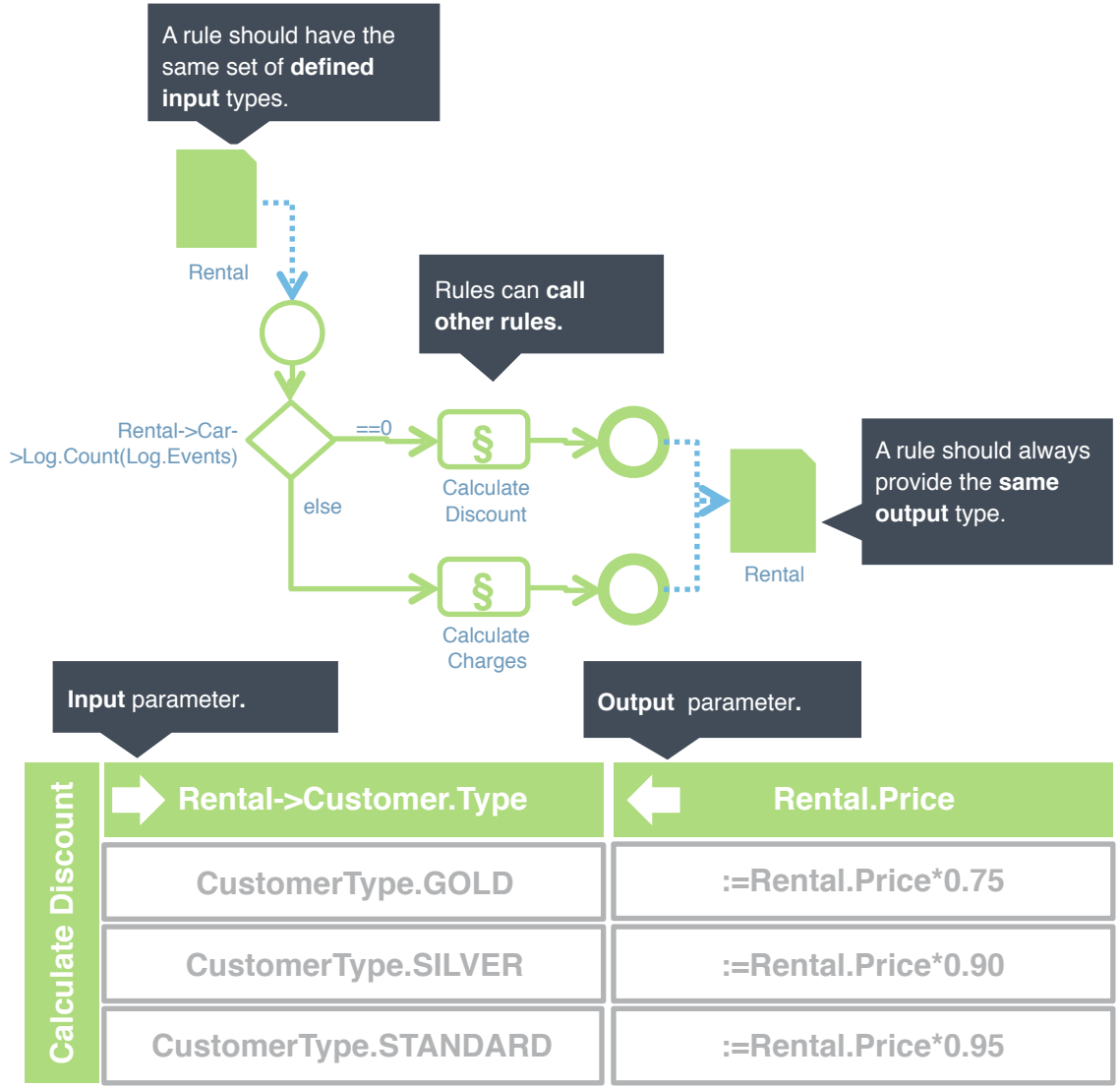
Business Rules work on business objects.

Business Rules usually come in two flavors that can be combined

- **Tree-based** (if—then—else style)
- **Tabular** (Excel style)

Business rules help in

- **Making processes flexible**, due to their different lifecycle
- **Centralizing & re-using business logic** in different processes or other parts of the application



USER INTERFACES

User Interface 1x1:

- Create a simple, repeating structure of core building blocks that can be easily arranged according to responsive layouts
  - **Use black/white schematics**
  - **Keep it simple**
  - **Focus on functionality**

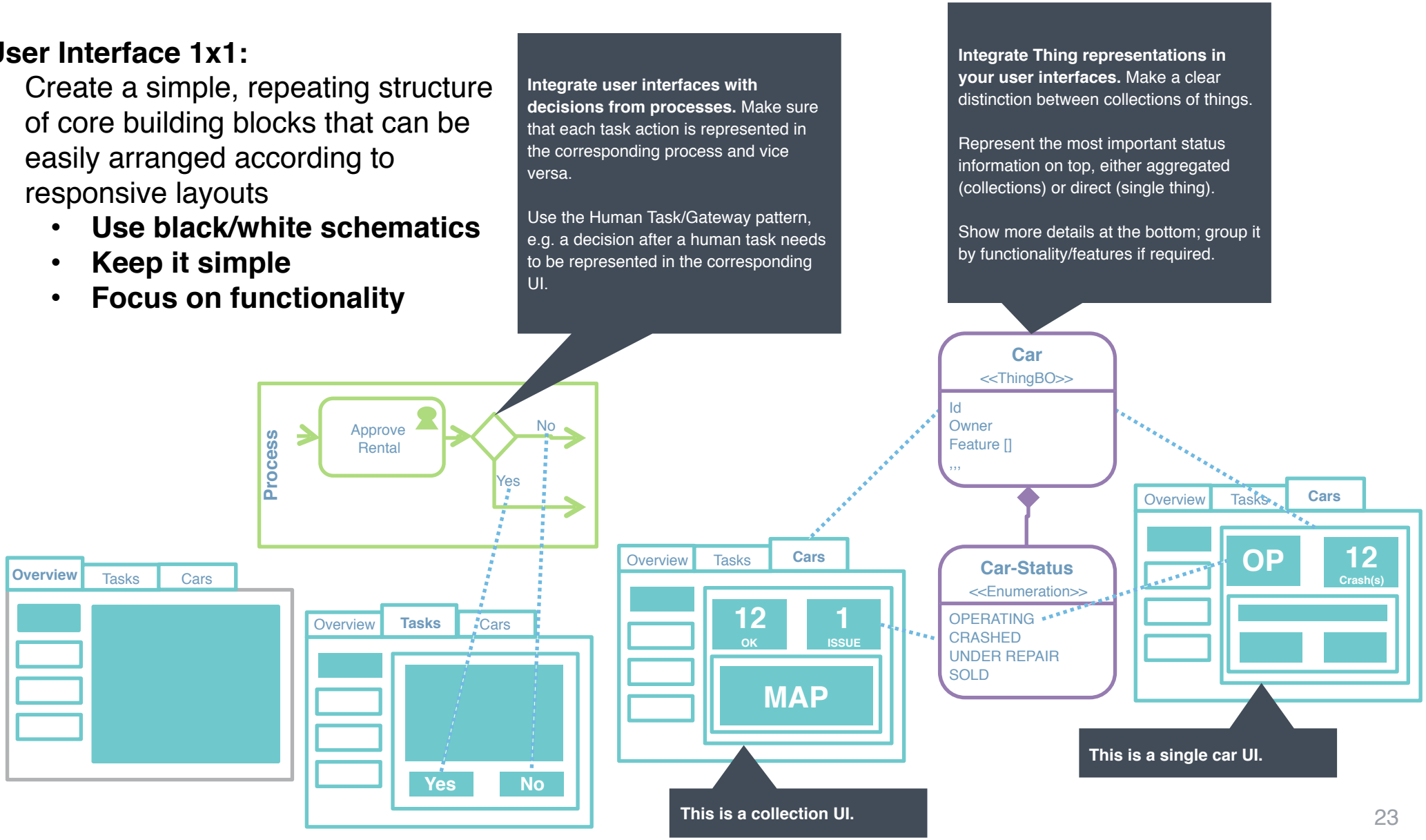
Integrate user interfaces with decisions from processes. Make sure that each task action is represented in the corresponding process and vice versa.

Use the Human Task/Gateway pattern, e.g. a decision after a human task needs to be represented in the corresponding UI.

Integrate Thing representations in your user interfaces. Make a clear distinction between collections of things.

Represent the most important status information on top, either aggregated (collections) or direct (single thing).

Show more details at the bottom; group it by functionality/features if required.



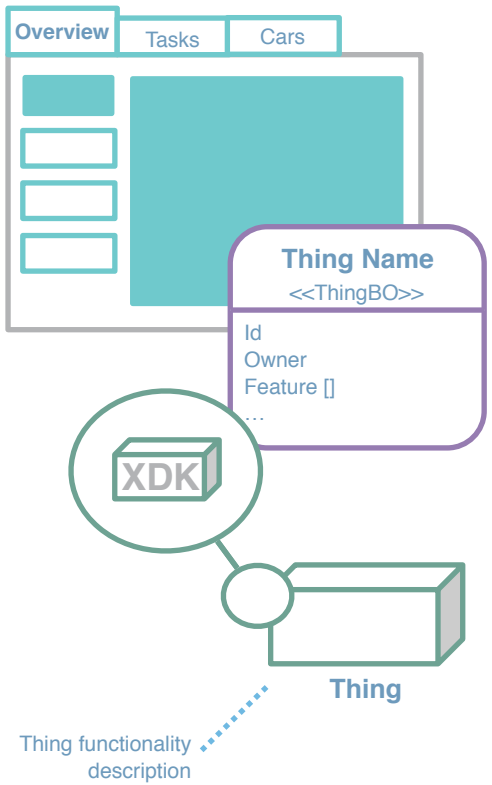
CONNECTIVITY

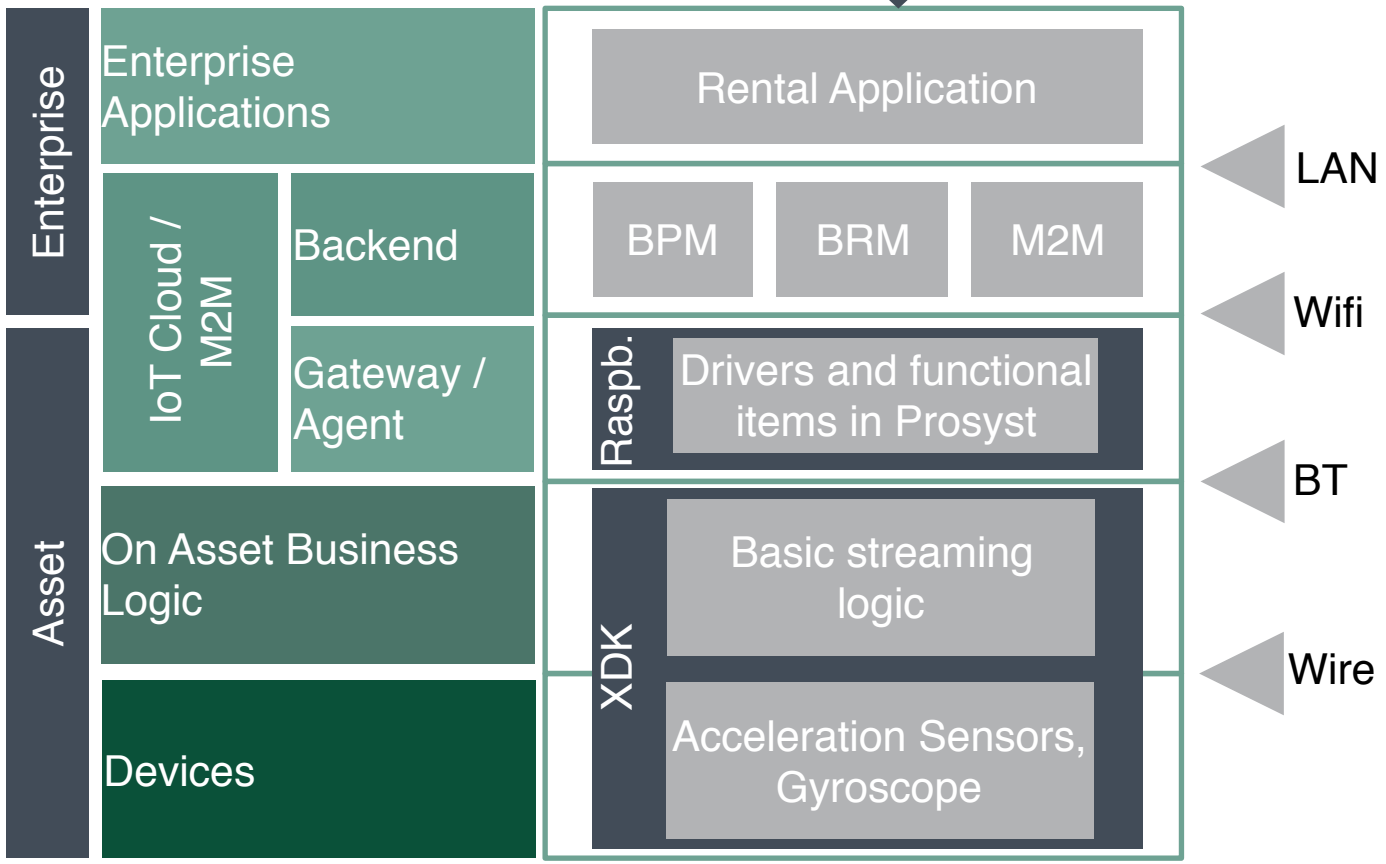
CONNECTIVITY

Asset Integration Architecture (AIA).
Describe the different connectivity layers your solution requires from the devices to the Enterprise application.

This is just an example AIA. Try to map your problem domain on the different layers (and remove them if not necessary). Also annotate the connectivity between the layers.

Find many more AIA examples throughout the Enterprise IoT book.

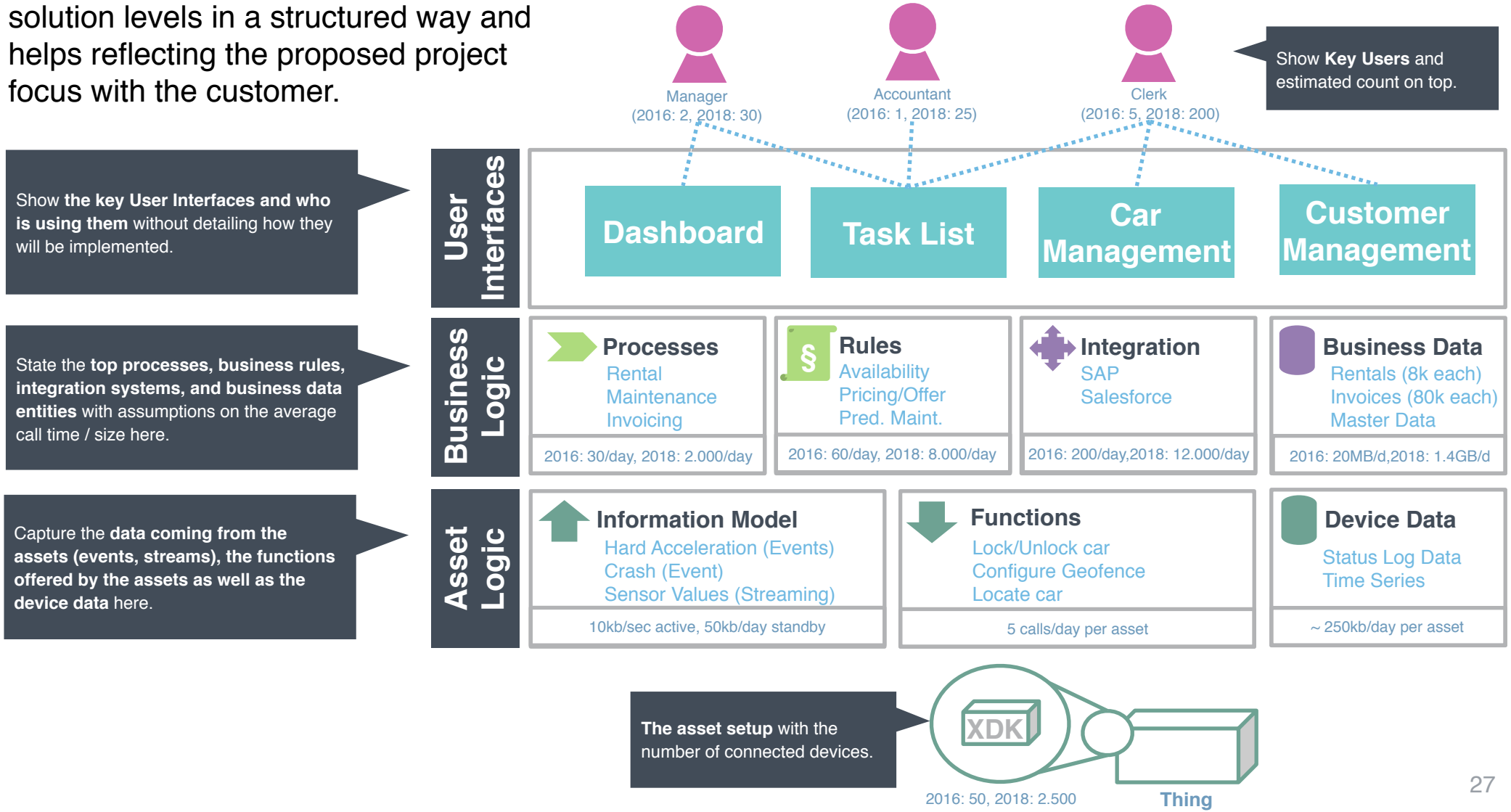




FINALIZE

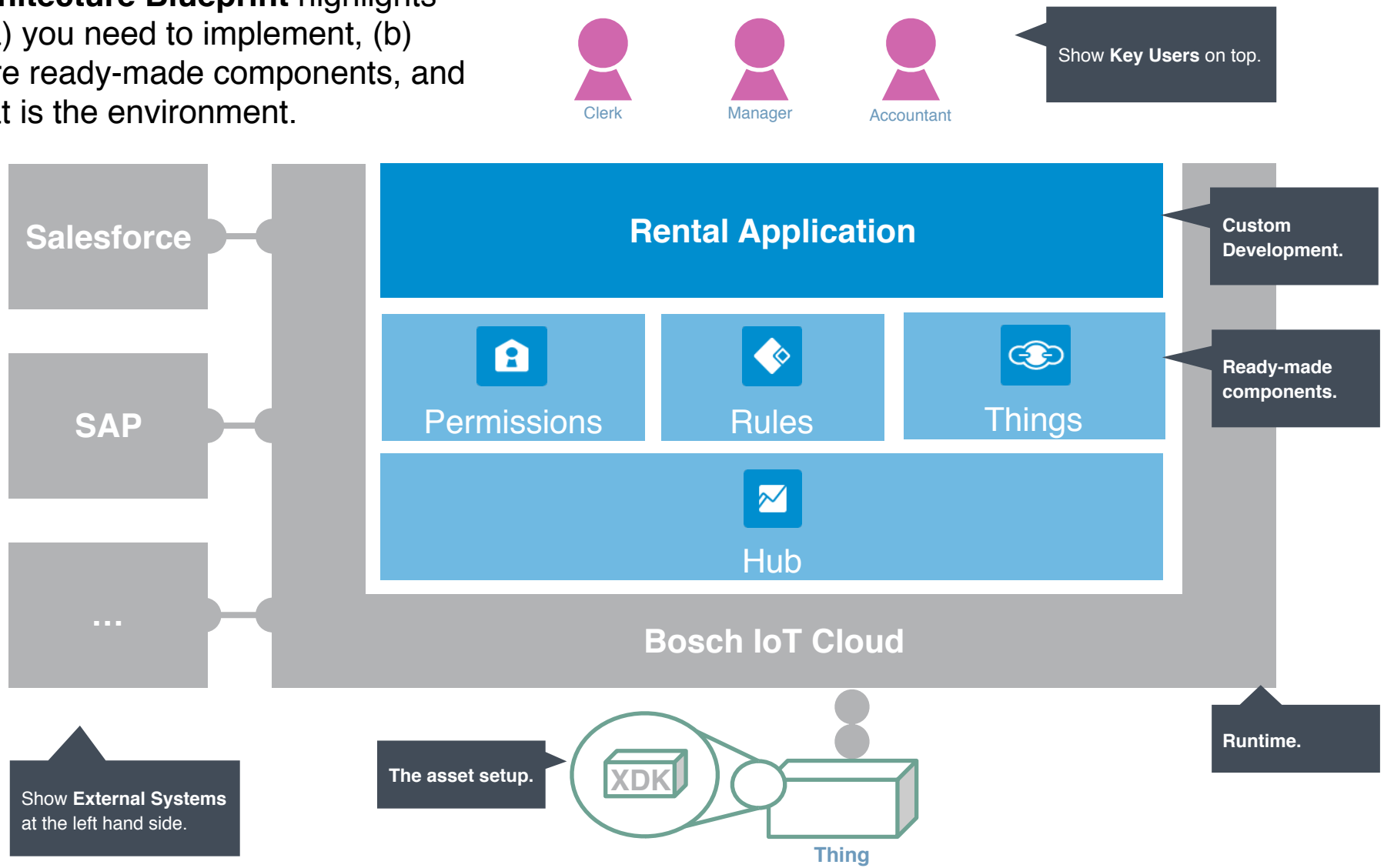
PROJECT SKETCH

A **Project Sketch** summarizes the key solution levels in a structured way and helps reflecting the proposed project focus with the customer.



ARCH. BLUEPRINT

An **Architecture Blueprint** highlights what (a) you need to implement, (b) what are ready-made components, and (c) what is the environment.



WRAP UP

Wrapping up for the final pitch. Make sure you have worked well with your customer to understand

- the problem at hand via a **Solution Draft**
- the **refined views** of the stakeholders & roles, the domain model, the AIA for connectivity, the processes and rules, as well as the user interfaces

Make sure that your customer finds his understanding reflected in your filled templates. Always let him review and iterate the results by his own.

Propose concrete next steps by using

- A **Project Sketch** to show the functionality covered
- An **Architecture Blueprint** to show how an IoT Suite can help

Online Resources:

- Find the Enterprise IoT book online with many more examples http://enterprise-iot.org
- Read the Bosch Blog on Internet of Things http://blog.bosch-si.com
- Describe your things in a standardized way http://www.eclipse.org/vorto/