Software Engineering 2 (e20)

First project presentation

Ekkart Kindler

DTU Compute
Department of Applied Mathematics and Computer Science

- Background
- Idea
- Organization

Live Road Assessment (LiRA) based on modern cars' sensors

Ekkart Kindler

DTU Compute
Department of Applied Mathematics and Computer Science

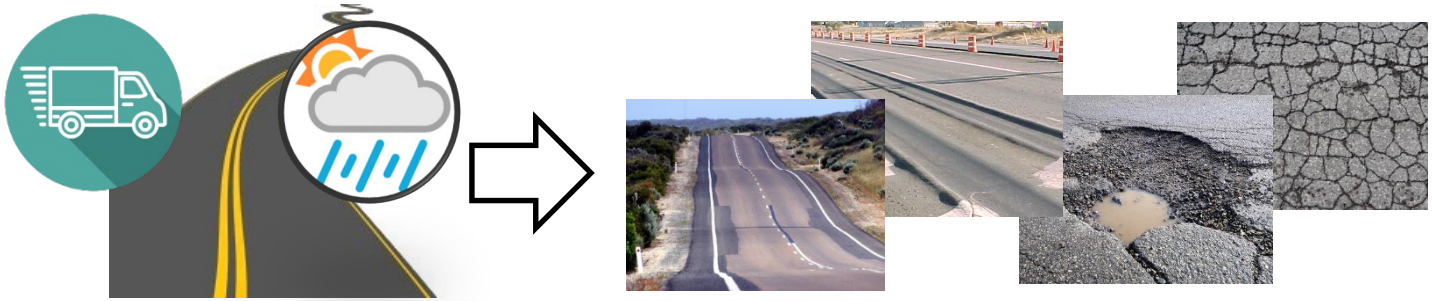
Innovation Fund Project (Grand Solutions)
Danish Road Authority (Vejdirektoratet, DRD)
DTU Byg
DTU Compute (SPE & CogSys)
Green Mobility
SWECO

Some slides "borrowed" from
Matteo Pettinari (Projektleader, DRD)

Background: LiRA

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

Roads make a crucial contribution to economic development and growth and bring **important** social benefits.



Background

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

Standard road measures have been developed to guarantee proper road conditions and to optimize maintenance strategies focusing on (DRD operational costs 5 million DKK per year – do not include Env. Emissions) :

- Safety
- Comfort
- Durability
- Environmental emissions (noise and CO_2)

Problems & Limitations

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

- Weather
- Costs
- Road geometry
- Not (always) objective
- Frequency

When road maintenance is started too late, maintenance costs increase exponentially with time!

LiRA: Main Idea

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

Can we find a more efficient and faster way to **monitor**, maintain and manage the roads?

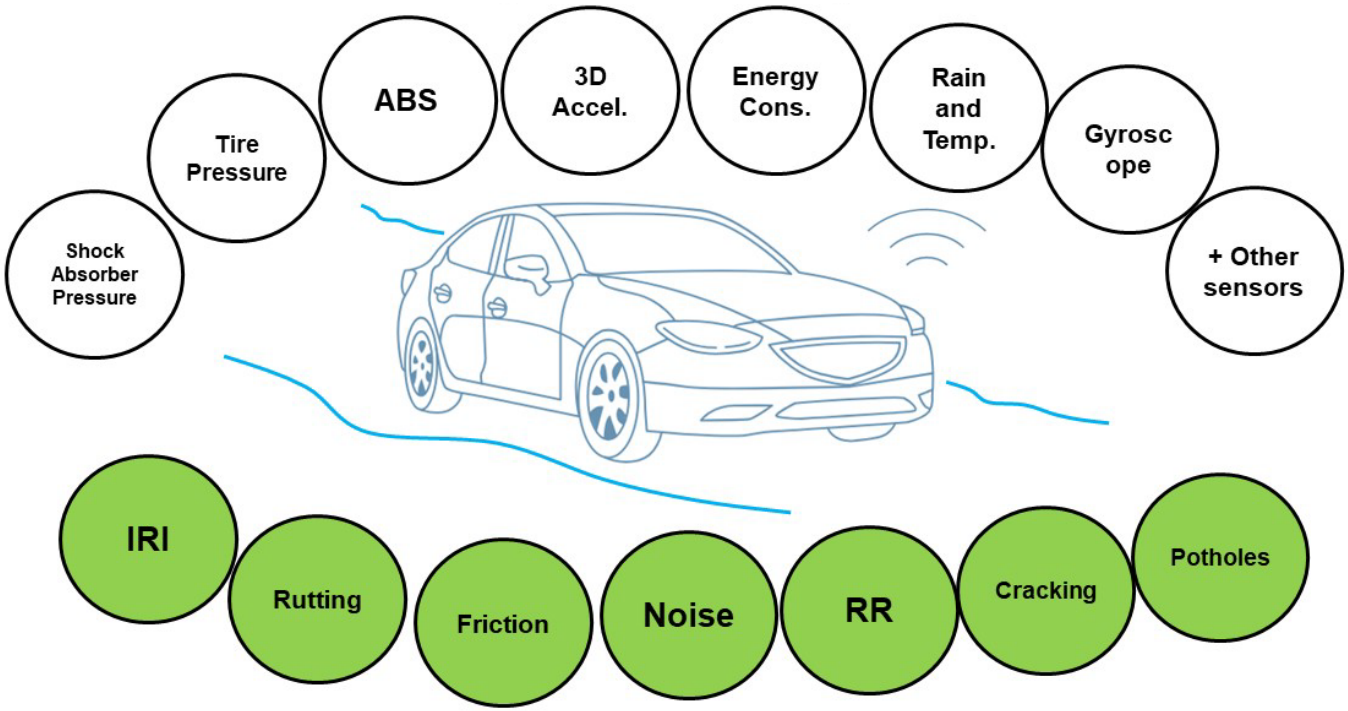
Modern cars are equipped with many sensors and can also provide further data including energy consumption.

Can car sensors data be used to measure road conditions?

Project idea

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

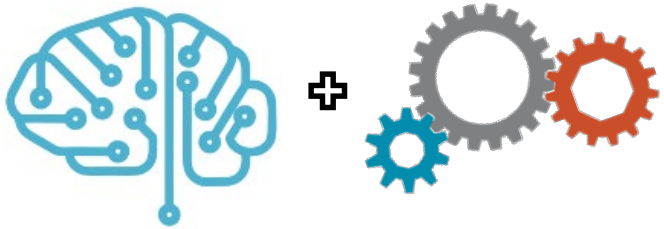
Car sensors (approx. 150 sensors)



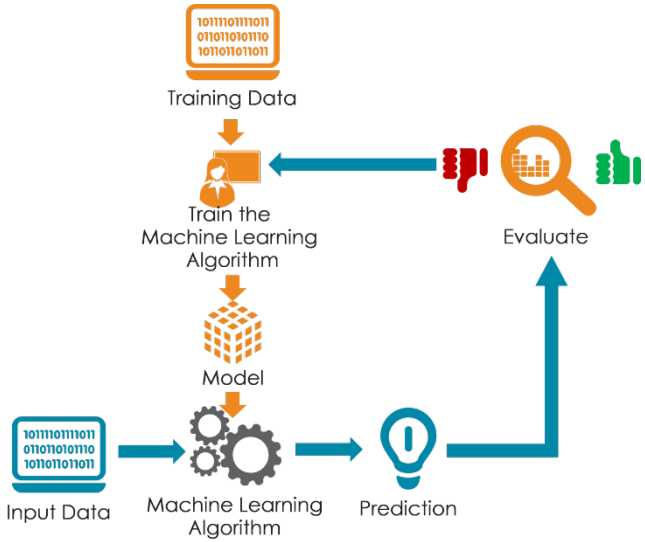
Road measures

Machine Learning

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

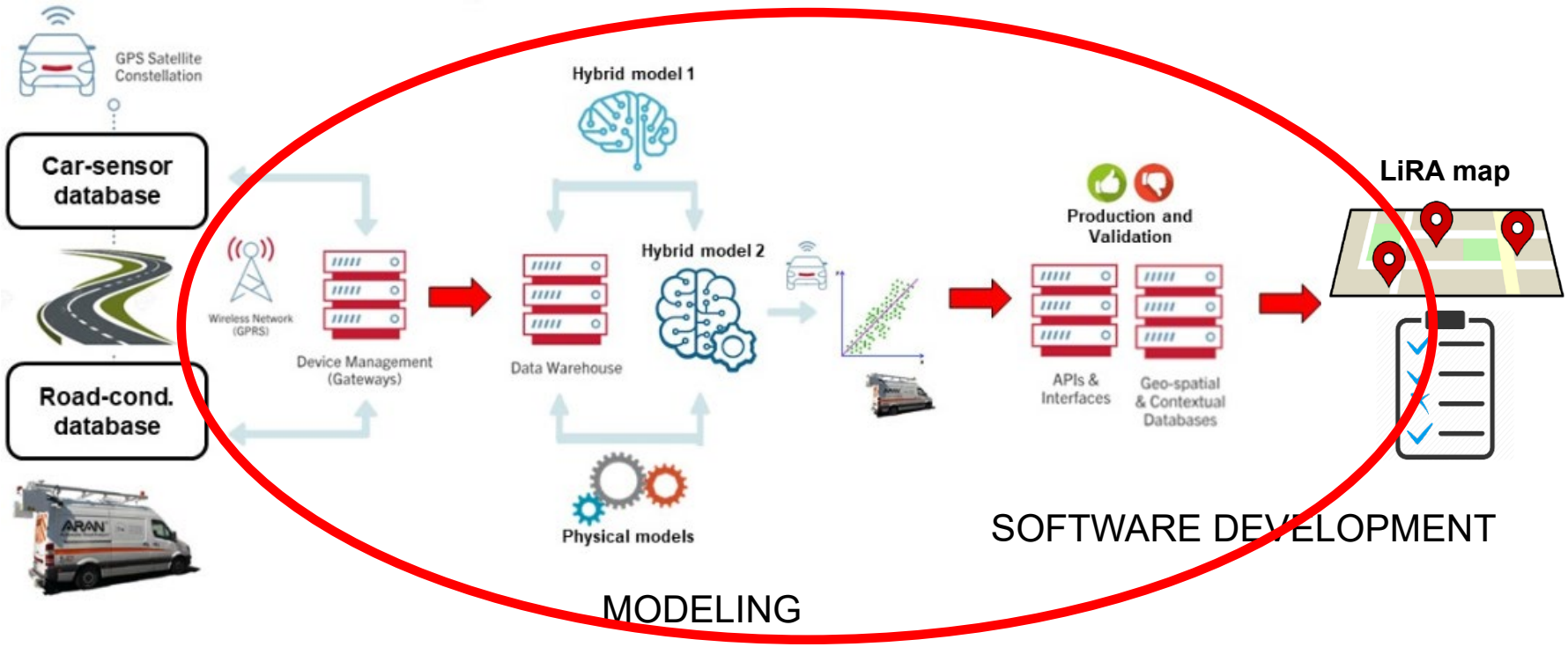


- Training (supported by physical models)
- Validation & Testing (Municipality of Copenhagen)



DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

Overview



Current State

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

- Implemented systems for collection and processing of data (car data as well as road data)
- Generic model for the collected data and road state data
- Implemented a system for visualizing road state data on top of maps (Open Street Map)
- Simple export of data for ML, but ML not yet integrated to this system
- Road maintenance processes not yet supported

Technologies

- Open Street Map (OSM) and friends
- Micro services (message Queues)
- C# and others
- …

More details will be presented on Friday and over the next few weeks!

LiRA: Material

- *LiRA Home page: http://lira-project.dk/*
- Jonathan Drud Bendsen: LiRA Map: A Cloud-based Geo-information System for Road Maintenance. BSc project 2020, https://findit.dtu.dk/en/catalog/2594682144
- *Markus Berthold*: Live Road Assessment based on modern cars: A prototypical geographic information system for road maintenance planning. MSc project 2020, https://findit.dtu.dk/en/catalog/2496058228
- More internal material will be online on CampusNet/Inside

- Background
- Idea
- Organization

Problem

- Currently, the system is working but consists of different parts that are not integrated yet
- The software is not too flexible, extensible and generic
- A proper user concept is missing
- Maintenance processes are not yet supported

Tasks

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

- Integrate core parts (data) and visualization
- Integrate them and make them more flexible, extensible and generic, including but not restricted to
  - Dynamically setting up pipelines for processing and pre-processing collected data
  - Adding mechanism for starting ML and obtaining the condition data (note that ML itself is out of scope)
- Visualization of existing car data on maps and accessing and filtering it
- Supporting a minimal road maintenance process
- …

Groups can suggest their own features, which then need to be discussed and prioritized with the project owner (→ planning game).

Out of scope

- Data collection from cars: You will be provided with a lot of data (raw data and road condition data), so that you can also do processing and pre-processing of these data without collecting it.
  If you want, you can emulate submission of data from cars (subject to planning game with Ekkart and Shahrzad), though.
- Machine Learning (ML) from road data (also here, you can emulate ML).
- Other map material than OSM

More Details

- More details will be presented in the next session on Friday and discussed over the next weeks

- Read up on LiRA (see material slide)
- Think about user stories
- Think about questions and unclear points!

- Background
- Idea
- Organization

Weekly Schedule (roughly)

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

	Mon	Tue	Wed	Thu	Fri
8-10		lecture			
10-12					
13-15					
15-17					project

lecture | tutorial | project

The group meetings on Friday (15-17) are mandatory! Each group will have their separate room (in building 210)

The tutorials

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

- The tutorials' topics cover some core technologies of the project
- Groups may make suggestions which information they would need and which topics should be covered

Project sessions

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

- Hands-on work
- Questions and Answers (Q&A) on the project
- Advice & consultation
- Feed back and comments on submissions

- Individual group meetings
  - for planning and organisation (including planning game)
  - work on the project
  - discussion of user stories

“Homework” to Friday

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

- Understand the project
  - Read up on project and technologies
  - Think of technologies
  - Think of and formulate user stories
  - What tasks could be covered by release 0
    (we will be playing a first, still informal, planning game on Friday)
- What could be your role in the group
  - Project leader and deputy
  - Which technical parts are you good at
    (programming, setting up GitHub/GitLab, Jenkins server, Webserver, mobile app)
  - How could you contribute to the project

Next Group Meeting

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

- Discuss the project
  - Break project down to user stories and tasks
  - Who takes which stories/tasks
    (remember that you need to work in pairs)
  - Organisation of TODOs and their status (e.g. Trello)
  - Tasks until Tuesday
  - Find a group leader and a deputy
  - Play the planning game (with Ekkart)

Ekkart (and sometimes Shahrzad, member of the LiRA project) will play the role of the project owner and an external consultant

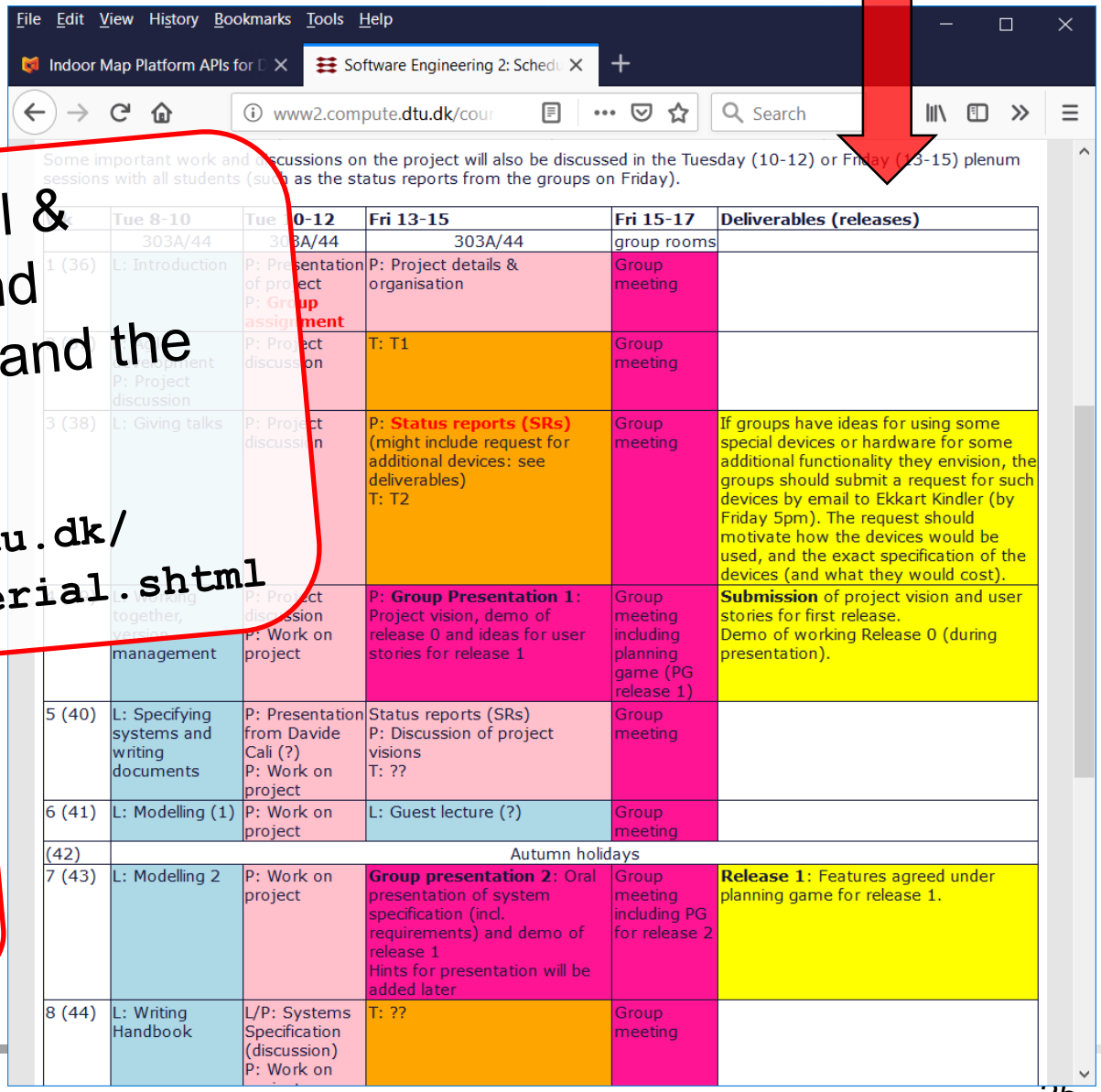
Releases/Submission

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

See web page (material & schedule) for details and deadlines for releases and the final submission:

`http://www2.compute.dtu.dk/courses/02162/e20/material.shtml`

Note: This is last year's web page. The pages for this year will be up soon.



Some important work and discussions on the project will also be discussed in the Tuesday (10-12) or Friday (13-15) plenum sessions with all students (such as the status reports from the groups on Friday).

	Tue 8-10	Tue 10-12	Fri 13-15	Fri 15-17	Deliverables (releases)
	303A/44	303A/44	303A/44	group rooms	
1 (36)	L: Introduction	P: Presentation of project P: **Group assignment**	P: Project details & organisation	Group meeting	
2 (37)	L: Requirement P: Project discussion	P: Project discussion	T: T1	Group meeting	
3 (38)	L: Giving talks	P: Project discussion	P: **Status reports (SRs)** (might include request for additional devices: see deliverables) T: T2	Group meeting	If groups have ideas for using some special devices or hardware for some additional functionality they envision, the groups should submit a request for such devices by email to Ekkart Kindler (by Friday 5pm). The request should motivate how the devices would be used, and the exact specification of the devices (and what they would cost).
4 (39)	L: Working together, version management	P: Project discussion P: Work on project	P: **Group Presentation 1**: Project vision, demo of release 0 and ideas for user stories for release 1	Group meeting including planning game (PG release 1)	**Submission** of project vision and user stories for first release. Demo of working Release 0 (during presentation).
5 (40)	L: Specifying systems and writing documents	P: Presentation from Davide Cali (?) P: Work on project	Status reports (SRs) P: Discussion of project visions T: ??	Group meeting	
6 (41)	L: Modelling (1)	P: Work on project	L: Guest lecture (?)	Group meeting	
(42)	Autumn holidays				
7 (43)	L: Modelling 2	P: Work on project	**Group presentation 2**: Oral presentation of system specification (incl. requirements) and demo of release 1 Hints for presentation will be added later	Group meeting including PG for release 2	**Release 1**: Features agreed under planning game for release 1.
8 (44)	L: Writing Handbook	L/P: Systems Specification (discussion) P: Work on	T: ??	Group meeting	

Shahrzad M. Pour, Postdoctoral Researcher at LiRA, Joined October 2019

- LiRA source code Specifications
- Architecture (hybridization of event-driven and microservice)
- LiRA Data Pipeline
- Overview on Source code

DTU Compute
Department of Applied Mathematics and Computer Science

Product Owner at LiRA Software Solution
email: shmp@dtu.dk

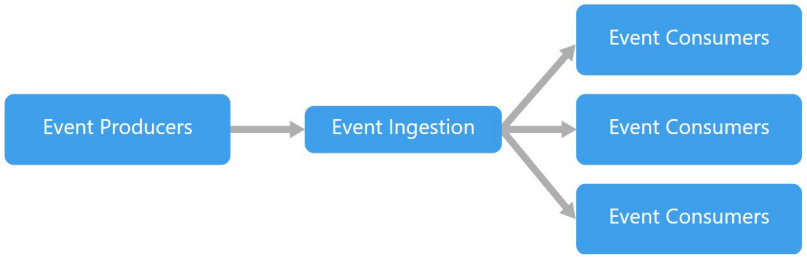
LiRA Source Code Spec

- Code environment: Visual Studio .NET 2019
- Programming language: C#
- Deployment Environment: Deployed as Dockers on an Ubuntu Server
- Architecture: Hybridization of Event-driven architecture, Microservice architecture
- Technologies: Rabbit MQ, Microsoft Entity Framework (As a Code-First Approach), Docker, RESTful Api

Hybridized Architecture

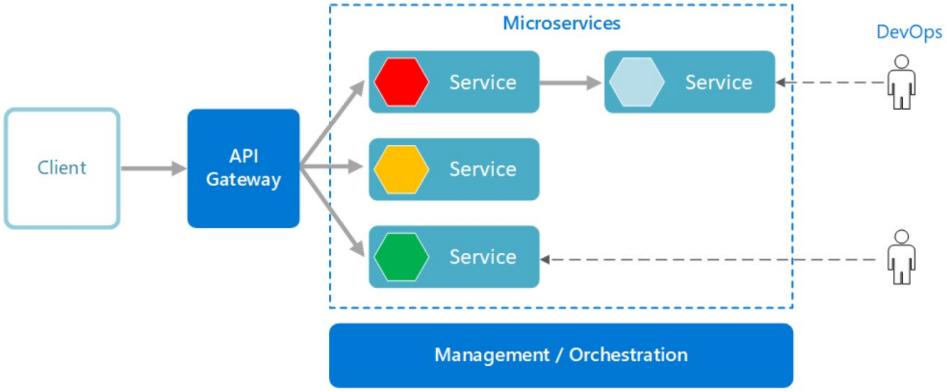
DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

- Event-Driven Architecture



http://docs.microsoft.com/da-dk/azure/architecture/guide/architecture-styles/event-driven

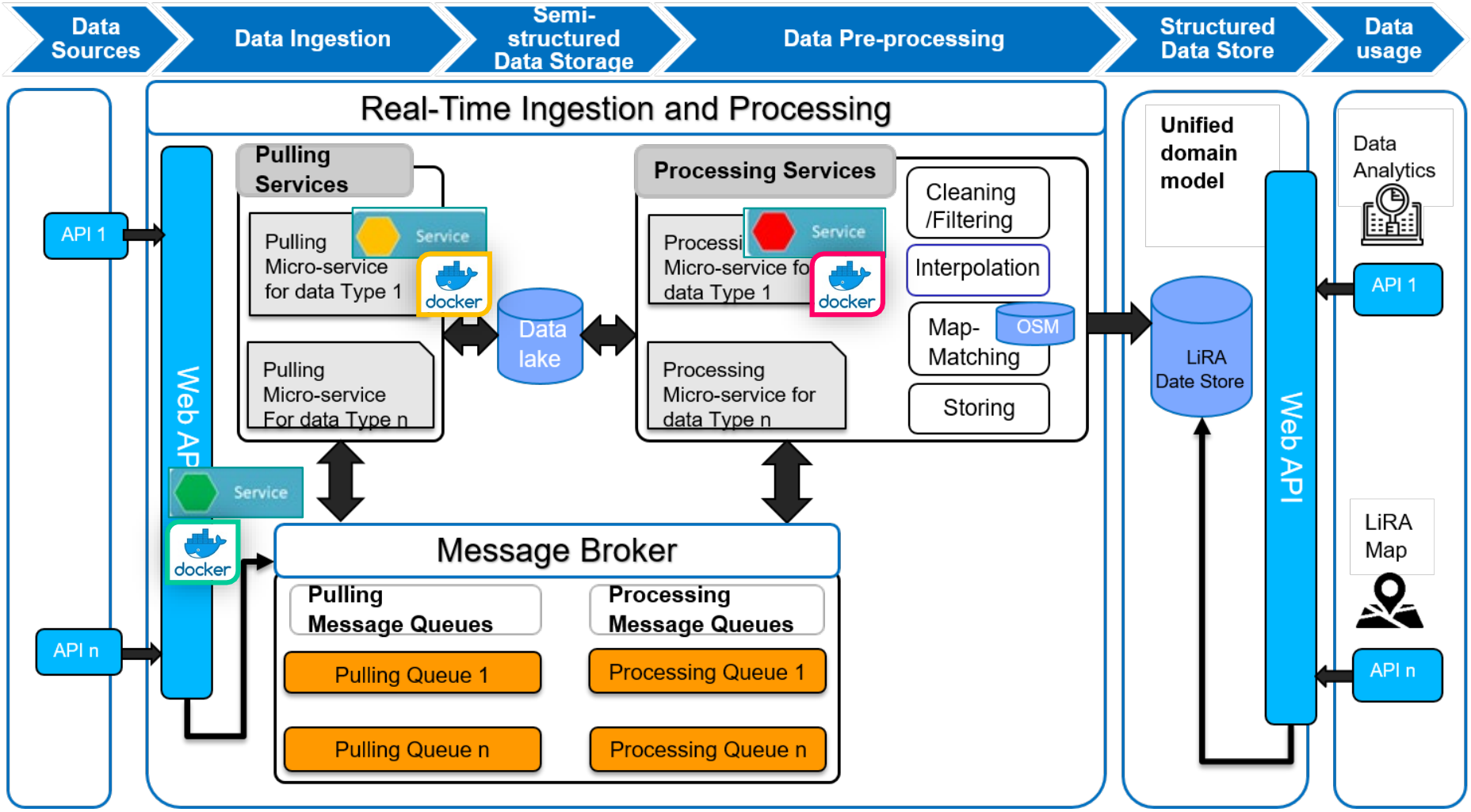
- Microservice Architecture



https://docs.microsoft.com/da-dk/azure/architecture/guide/architecture-styles/microservices

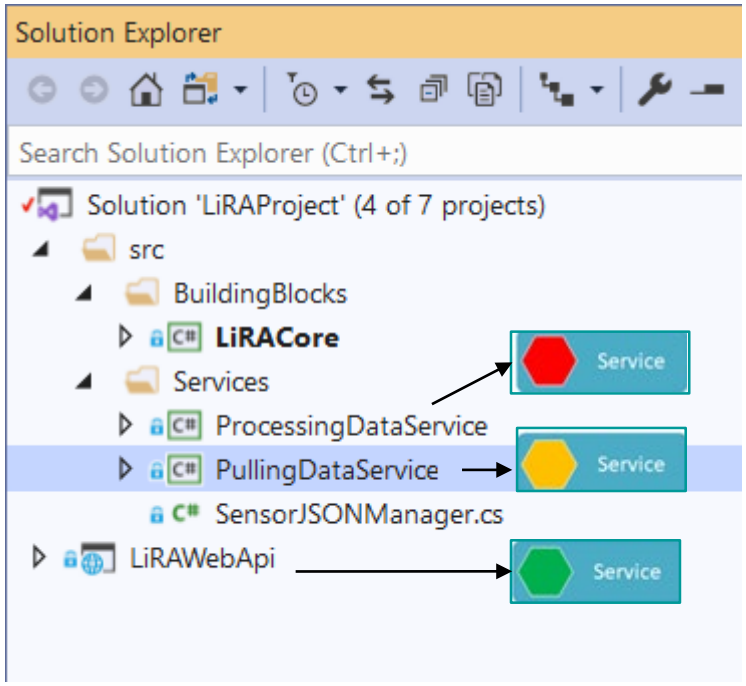
LiRA Data Pipeline

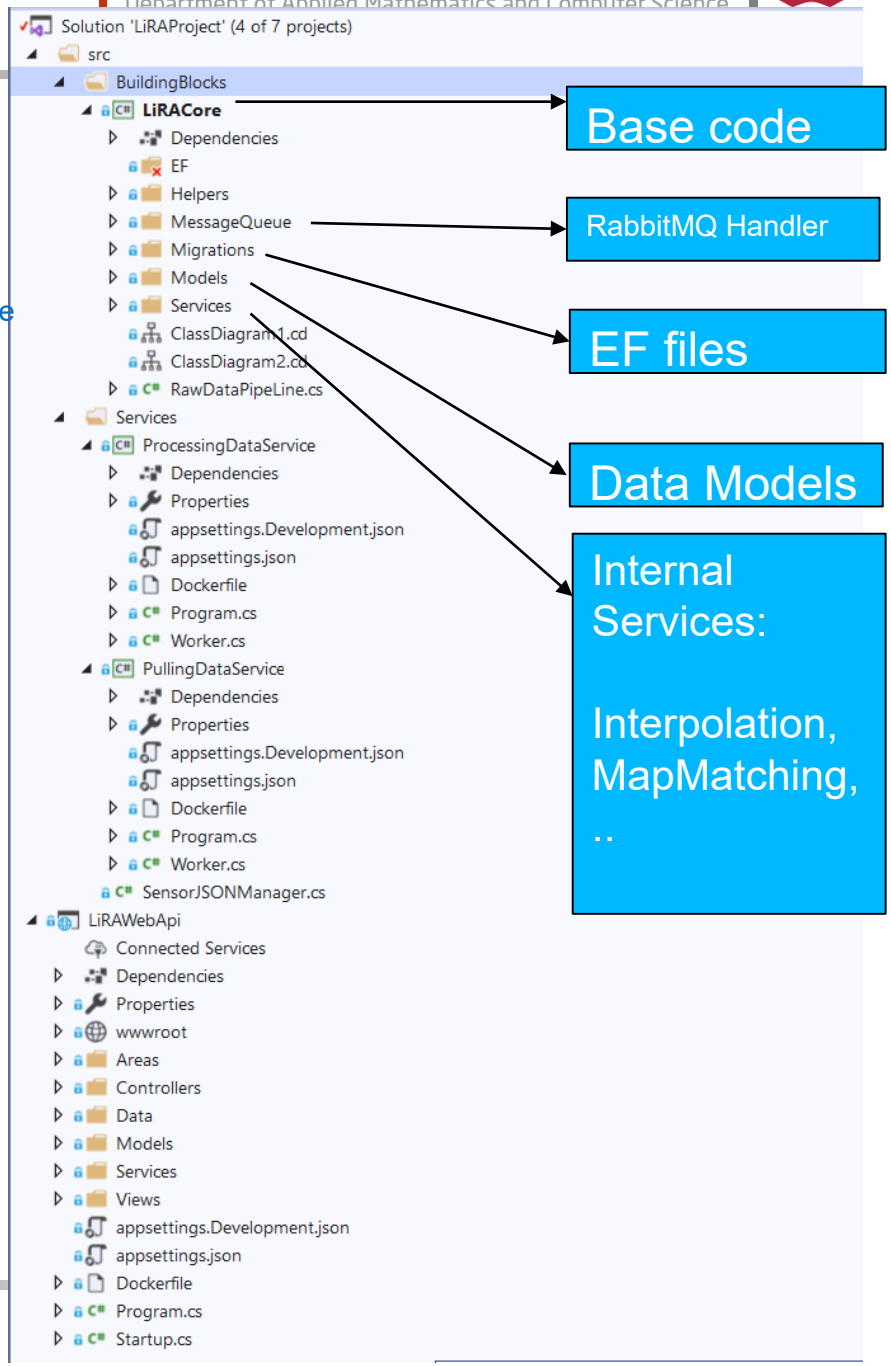
DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler



Overview of Source code

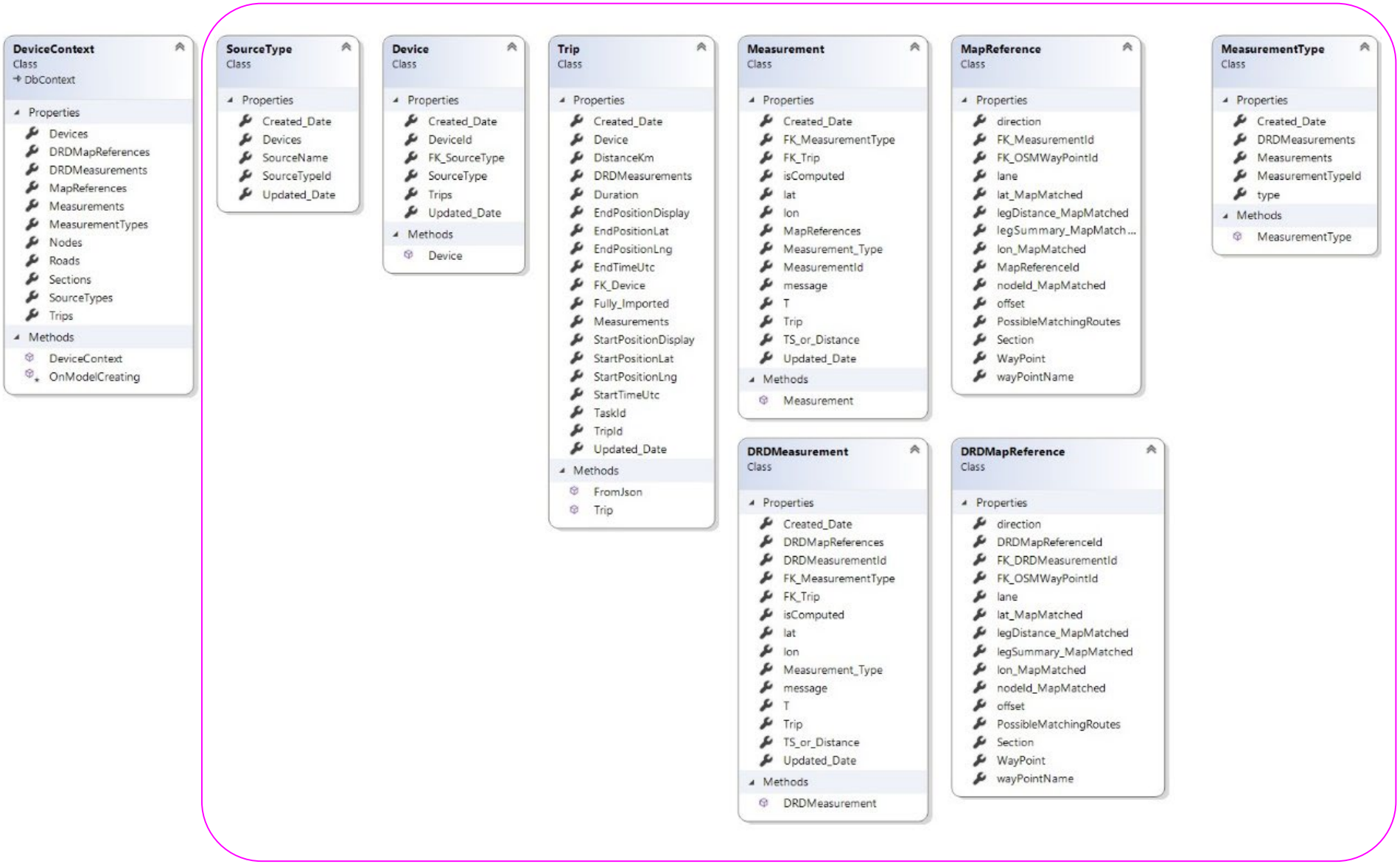
- Net solution contains Four .Net projects:
  - "LiRACore" project: Data models and common/helper classes.
  - "ProcessingDataService" project: Data processing microservice
  - "PullingDataService": Data pulling microservice
  - "LiRAWebApi": Notification receiver





Database Schema

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler



Install Docker Engine on Ubuntu

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

- Prerequisites
  - Ubuntu Focal 20.04 (LTS)
  - Ubuntu Bionic 18.04 (LTS)
  - Ubuntu Xenial 16.04 (LTS)

    64-bit
- Uninstall old versions (docker, docker.io, or docker-engine)
  - Verify by apt-get command
- Installation methods
  - Install using the repository
    - ease of installation and task upgrading
  - Install from a package (DEB)
    - useful for air-gapped systems with no internet access
  - Convenience Scripts
    - testing and development environment

Docker Installation

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

1. Install using the repository
   - SET UP THE REPOSITORY
     - Update the apt Package index and install packages
     - Add Docker's official GPG key
       - Key will be added from the download.docker.com page
       - Key can be verified
     - Set up the stable repository
   - INSTALL DOCKER ENGINE
     - Update the apt package index, and install the *latest version* of Docker Engine
     - To install a *specific version* of Docker Engine
       - List the available version
       - Install specific version
       - Verify Docker Engine is installed properly
   - UPGRADE DOCKER ENGINE

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

2. Install from a Package Manually

- INSTALLATION
  - Download the .deb file for your release (Ubuntu version)
  - Install Docker Engine
  - Verify the Docker Engine is installed
- UPGRADE
  - download the newer package file and repeat the installation procedure, pointing to the new
- Reference: https://docs.docker.com/engine/install/ubuntu/