Conception and Implementation of a robocup@Work refereebox

Kai Seidensticker, robOTTO

Requirements

- robcup@Work
- Referee Box Today
- Our Goals

Conception

- Overview
- Visual Design
- Feature

Timeline

- Open Questions
- Milestones

Future Work

- framework to create applications
- better design for frontend
- Morph
- eaieser deployment
  - use launchpad or reprepro

Conception and Implementation of a robocup@Work refereebox

Kai Seidensticker, robOTTO

Conception and Implementation of a robocup@Work refereebox

Kai Seidensticker, robOTTO

Requirements

- robcup@Work
- Referee Box Today
- Our Goals

Conception

- Overview
- Visual Design
- Feature

Timeline

- Open Questions
- Milestones

Future Work

- framework to create applications
- better design for frontend
- Morph
- eaieser deployment
  - use launchpad or reprepro

Requirements

robcup@Work

Referee Box Today

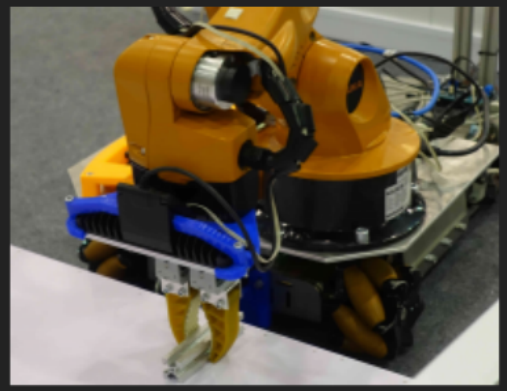
Our Goals

robOTTO

robocup@Work

- offical major league
- 5 -10 Team around the World
- a mobile manipulator solve Task in a miniatur Fabric
- technical Background
  - kinematic for manipulator
  - localisation
  - computer vision
- Focus:
  - Industry 4.0
  - Future Fab
  - Automation Technology

Kuka Youbot



robocup@Work

Requirements

robcup@Work

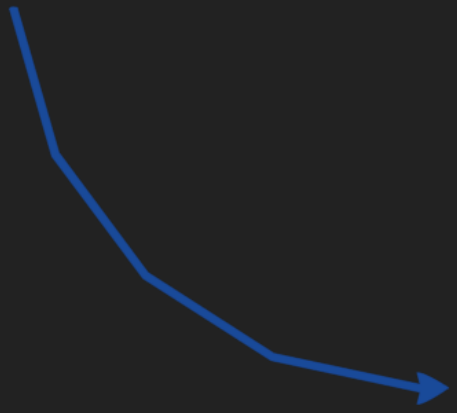
Referee Box Today

Our Goals

robOTTO

Referee Box Today

- two various referee box implementation
- no standardization
- no support for M2M communication
- no support features for referees
- pen and paper taskgeneration
- no visualisation for teams und audience



- the referee system isn't enable for a industry 4.0
- no benefit for referees, teams and league

Requirements

robcup@Work

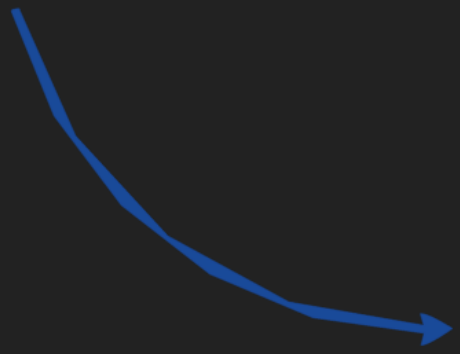
Referee Box Today

Our Goals

robOTTO

Our Goals

- one referee system for whole league
- a robocup proof solution
- uniform standards
- more referee support
- device integration (belt, roundtable)
- M2M communication
- visual support for audience



- a system for a industrial industry 4.0
- less work for the referees
- increasing the attractiveness of the league

Conception and Implementation of a robocup@Work refereebox

Kai Seidensticker, robOTTO

Requirements

- robcup@Work
- Referee Box Today
- Our Goals

Conception

- Overview
- Visual Design
- Feature

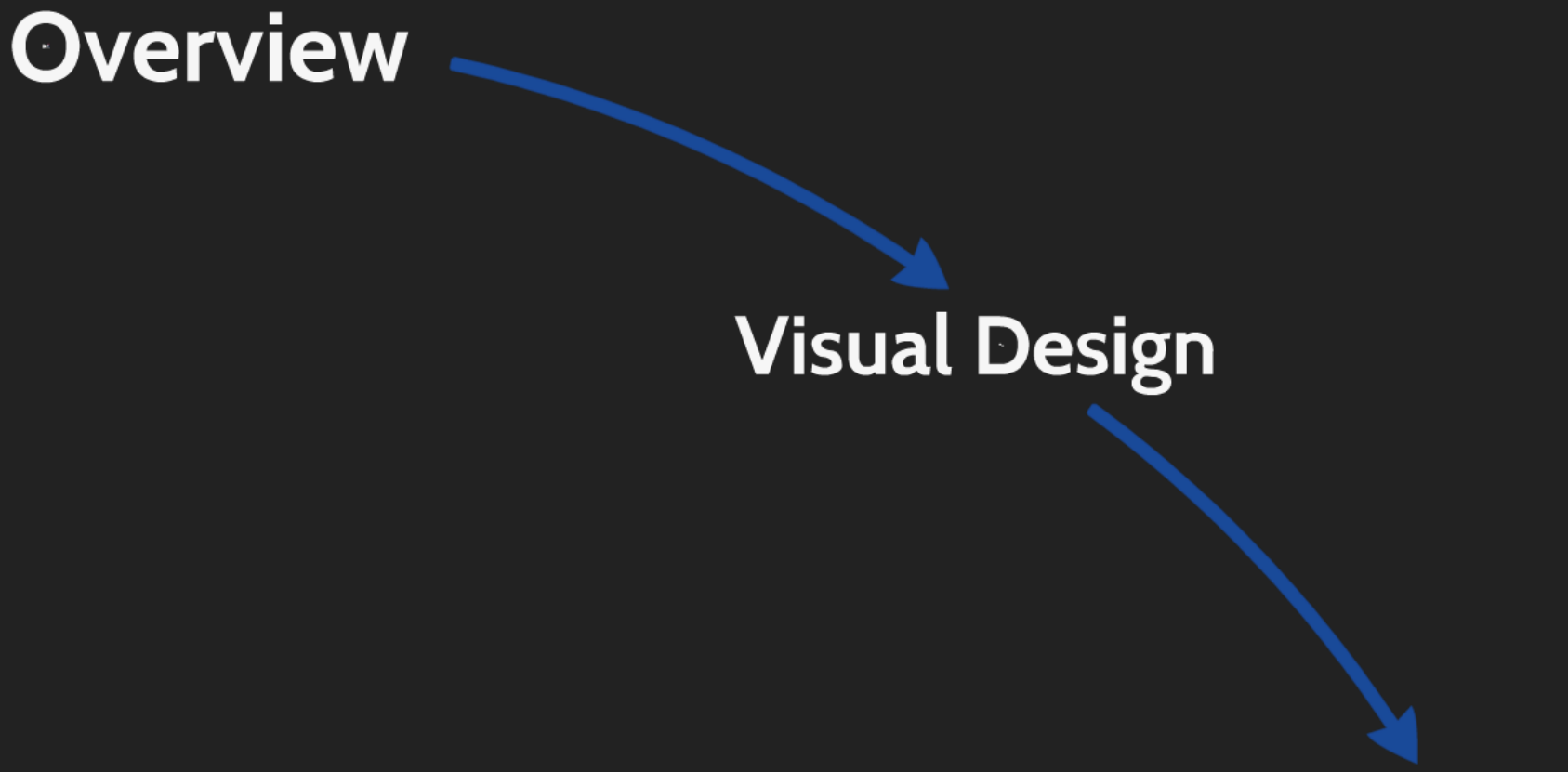
Timeline

- Open Questions
- Milestones

Future Work

- framework to create applications
- better design for frontend
- Morph
- eaieser deployment
  - use launchpad or reprepro

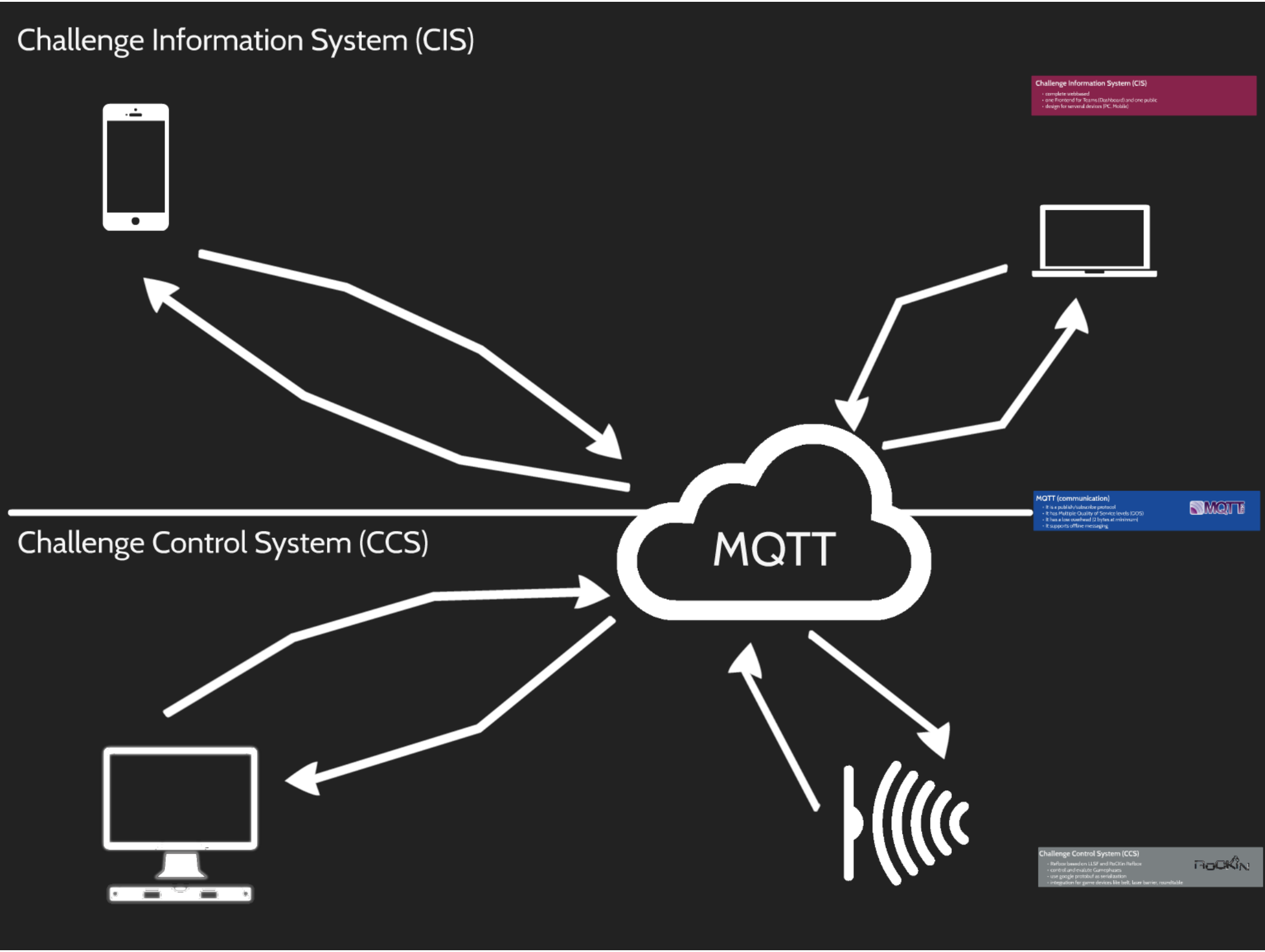
Conception



OTTO VON GUERICKE
UNIVERSITÄT
MAGDEBURG

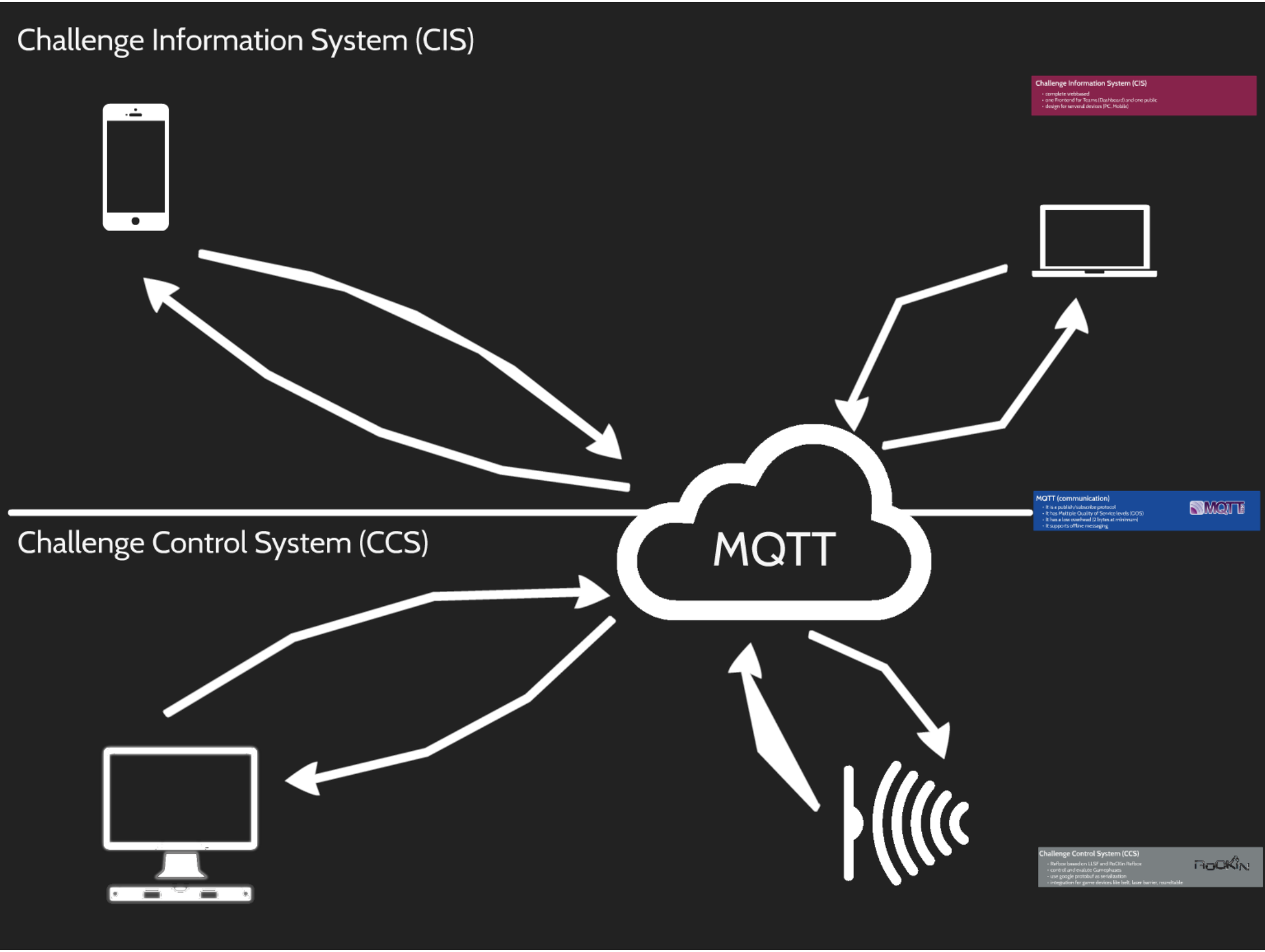
INF

robOTTO


Challenge Information System (CIS)
Challenge Information System (CIS)
• complete webbased
• one Frontend for Teams (Dashboard) and one public
• design for serveral devices (PC, Mobile)
MQTT (communication)
• It is a publish/subscribe protocol
• It has Multiple Quality of Service levels (QOS)
• It has a low overhead (2 bytes at minimum)
• It supports offline messaging
MQTT
Challenge Control System (CCS)
Challenge Control System (CCS)
• Refbox based on LLSF and RoCKIn Refbox
• control and evalute Gamephases
• use google protobuf as serialization
• integration for game devices like belt, laser barrier, roundtable

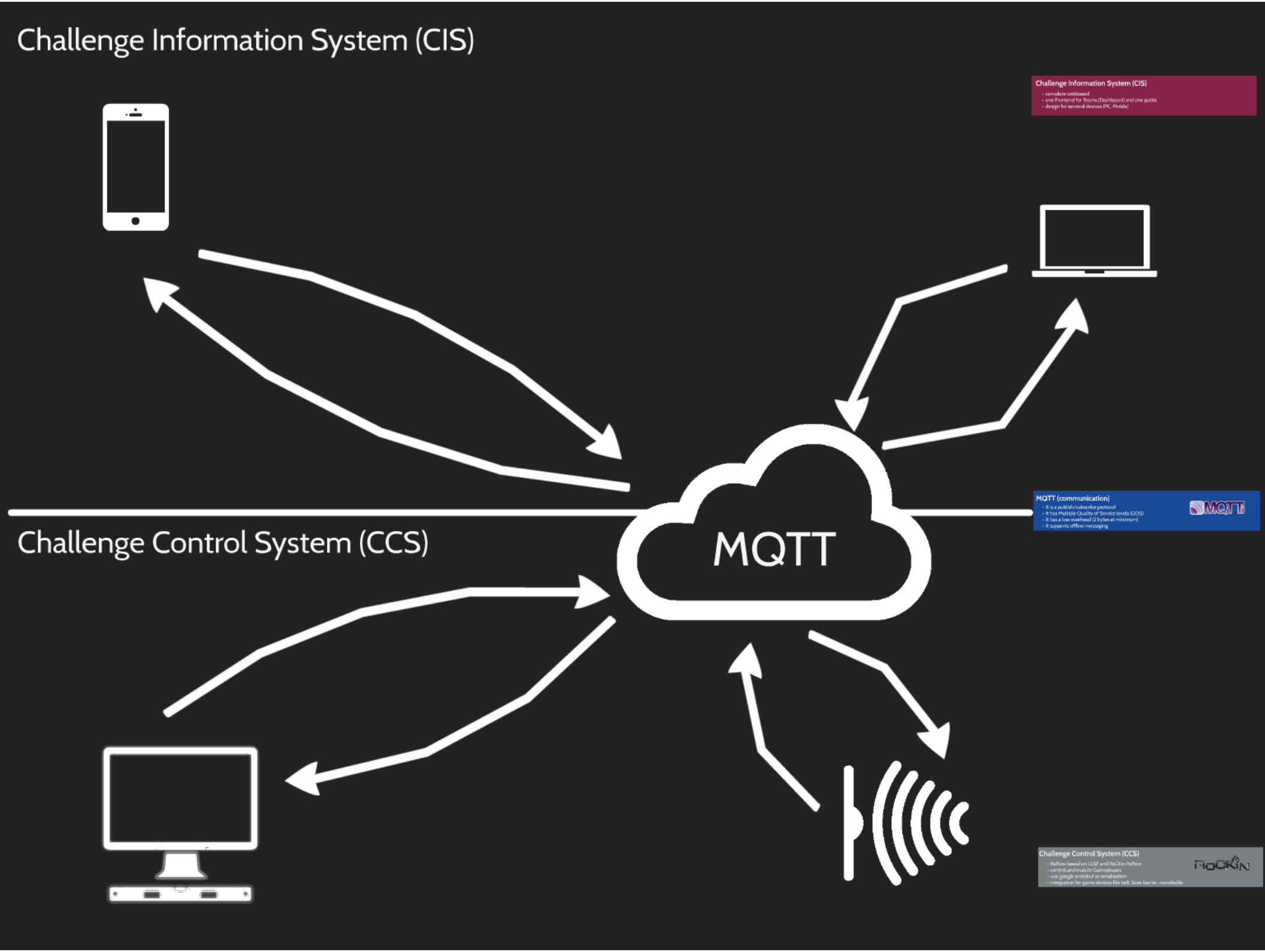
Challenge Control System (CCS)

- Refbox based on LLSF and RoCKin Refbox
- control and evalute Gamephases
- use google protobuf as serialization
- integration for game devices like belt, laser barrier, roundtable


Challenge Information System (CIS)
Challenge Information System (CIS)
• complete webbased
• one Frontend for Teams (Dashboard) and one public
• design for serveral devices (PC, Mobile)
MQTT (communication)
• It is a publish/subscribe protocol
• It has Multiple Quality of Service levels (QOS)
• It has a low overhead (2 bytes at minimum)
• It supports offline messaging
MQTT
Challenge Control System (CCS)
Challenge Control System (CCS)
• Refbox based on LLSF and RoCKIn Refbox
• control and evalute Gamephases
• use google protobuf as serialization
• integration for game devices like belt, laser barrier, roundtable

MQTT (communication)

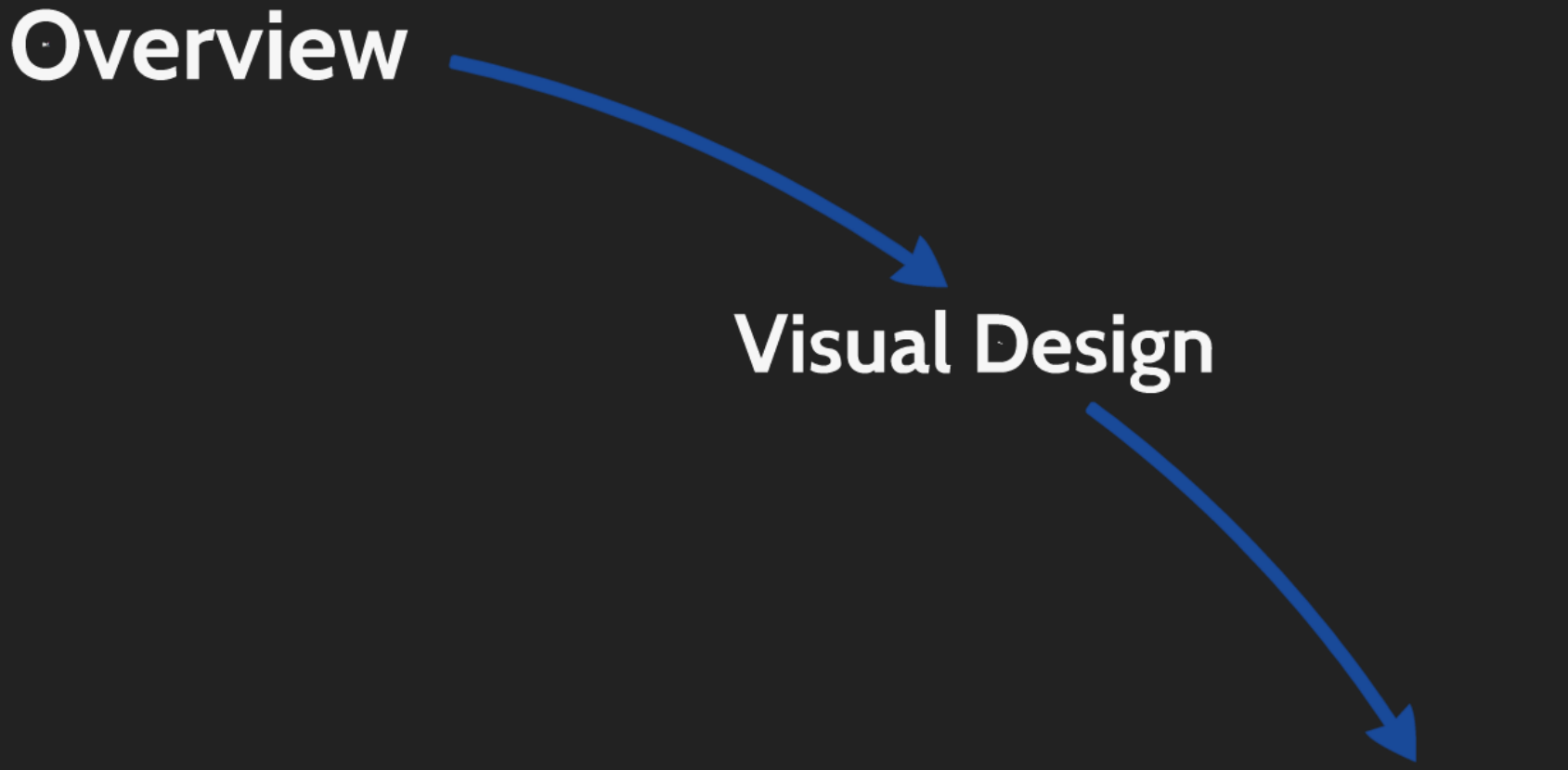
- It is a publish/subscribe protocol
- It has Multiple Quality of Service levels (QOS)
- It has a low overhead (2 bytes at minimum)
- It supports offline messaging


Challenge Information System (CIS)
Challenge Information System (CIS)
• complete webbased
• one Frontend for Teams (Dashboard) and one public
• design for serveral devices (PC, Mobile)
MQTT (communication)
• It is a publish/subscribe protocol
• It has Multiple Quality of Service levels (QOS)
• It has a low overhead (2 bytes at minimum)
• It supports offline messaging
MQTT
Challenge Control System (CCS)
Challenge Control System (CCS)
• Refbox based on LLSF and RoCKIn Refbox
• control and evalute Gamephases
• use google protobuf as serialization
• integration for game devices like belt, laser barrier, roundtable

Challenge Information System (CIS)

- complete webbased
- one Frontend for Teams (Dashboard) and one public
- design for serveral devices (PC, Mobile)

Conception

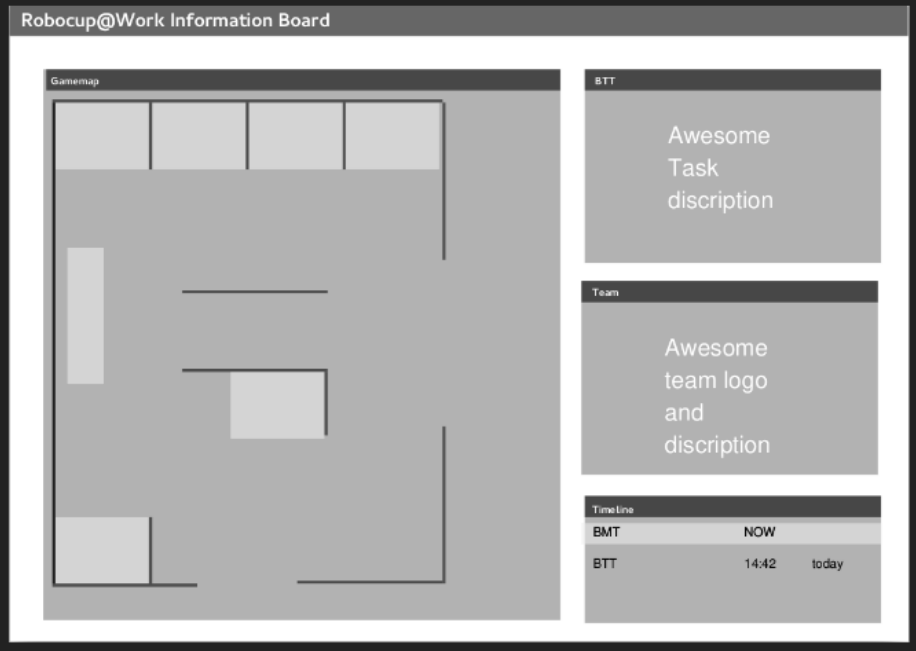


OTTO VON GUERICKE
UNIVERSITÄT
MAGDEBURG

INF

robOTTO

Team Dashboard, (mock version 0.7)



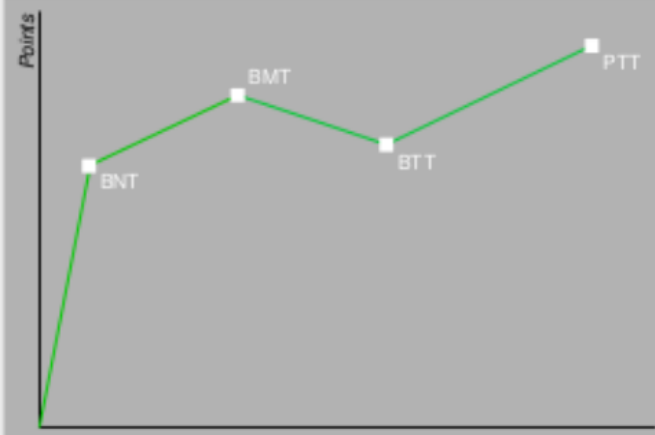
Public Dashboard, (mock version 0.7)

Dashboard

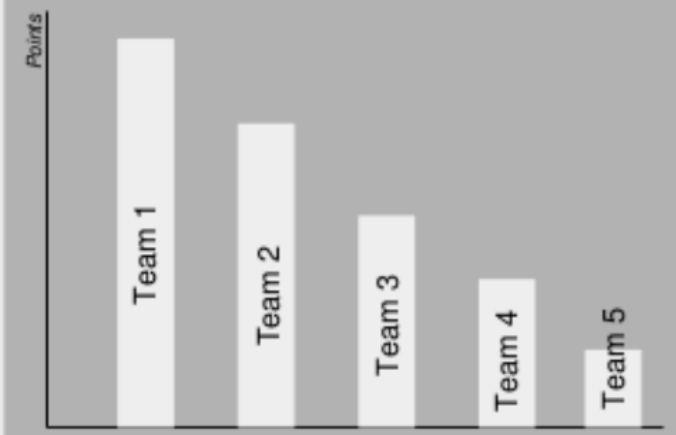
- Overview
- Tasks
- Intercom

Team Foobar

Own Results



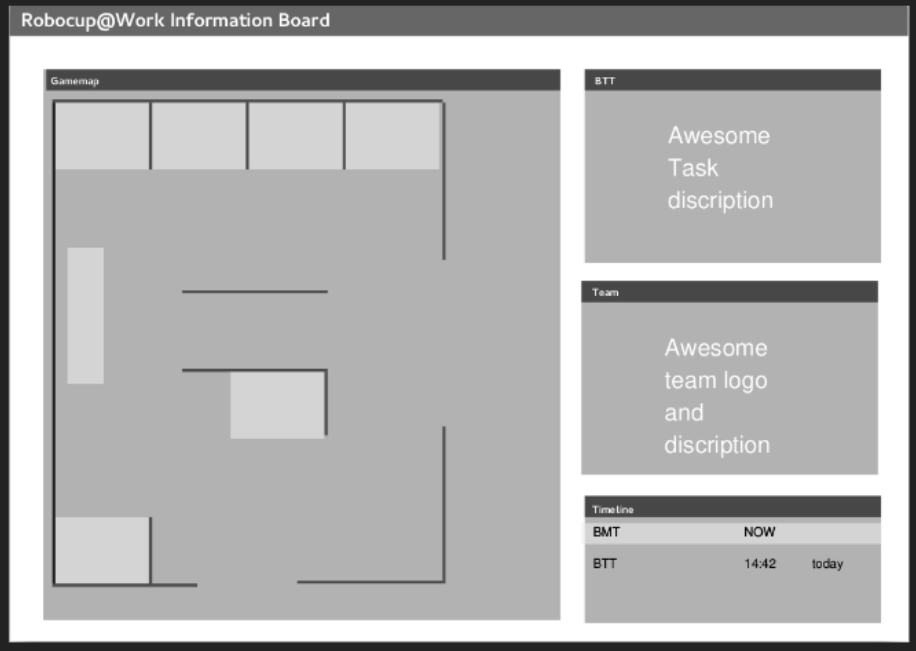
Result overall



Timeline

Team Leader Meeting	20:15	today
BMT	13:37	tomorrow

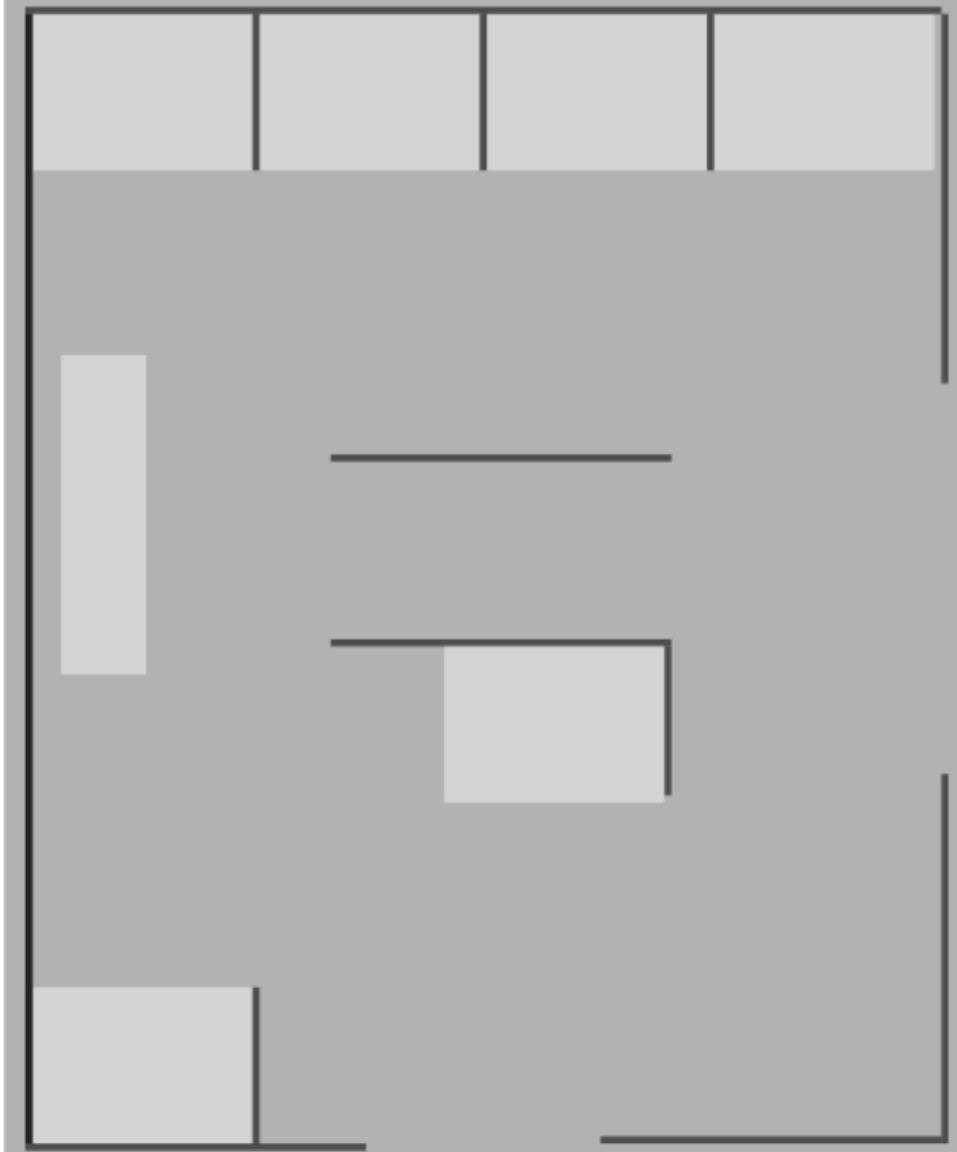
Team Dashboard, (mock version 0.7)



Public Dashboard, (mock version 0.7)

Robocup@Work Information Board

Gamemap



BTT

Awesome Task discription

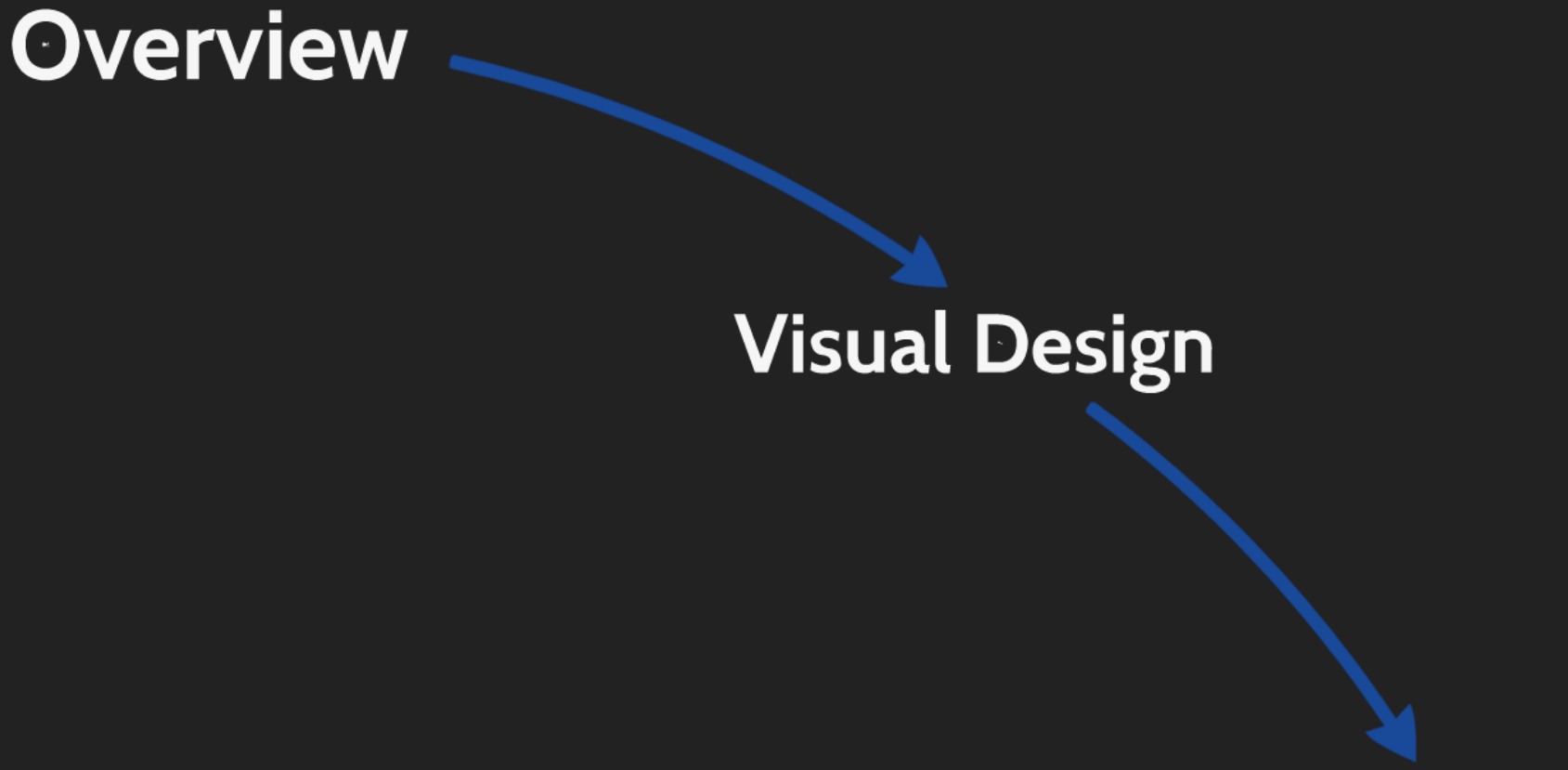
Team

Awesome team logo and discription

Timeline

BMT	NOW	
BTT	14:42	today

Conception



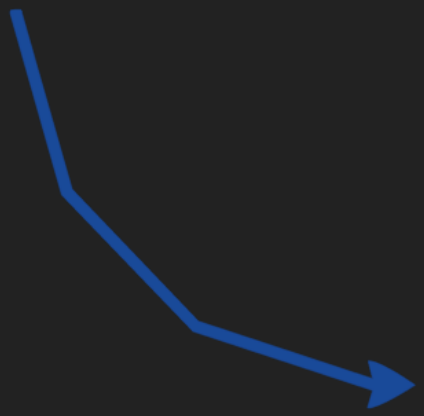
OTTO VON GUERICKE
UNIVERSITÄT
MAGDEBURG

INF

robOTTO

Feature

- automatization for many referee task
- M2M for devices and youbot
- automatical task generation
- visual support for audience
- better communication for teams
- workflow between mobile manipulator and model fab



communication system for Industry 4.0

Conception and Implementation of a robocup@Work refereebox

Kai Seidensticker, robOTTO

Requirements

- robcup@Work
- Referee Box Today
- Our Goals

Conception

- Overview
- Visual Design
- Feature

Timeline

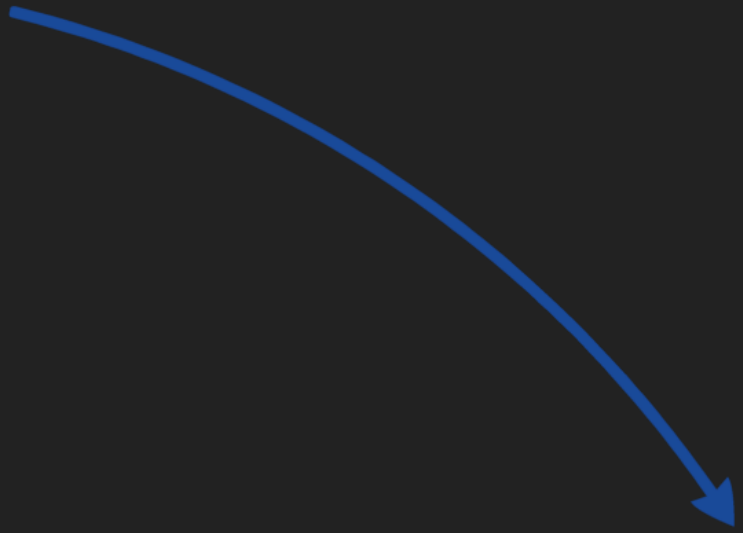
- Open Questions
- Milestones

Future Work

- framework to create applications
- better design for frontend
- Morph
- eaieser deployment
  - use launchpad or reprepro

Timeline

Open Questions

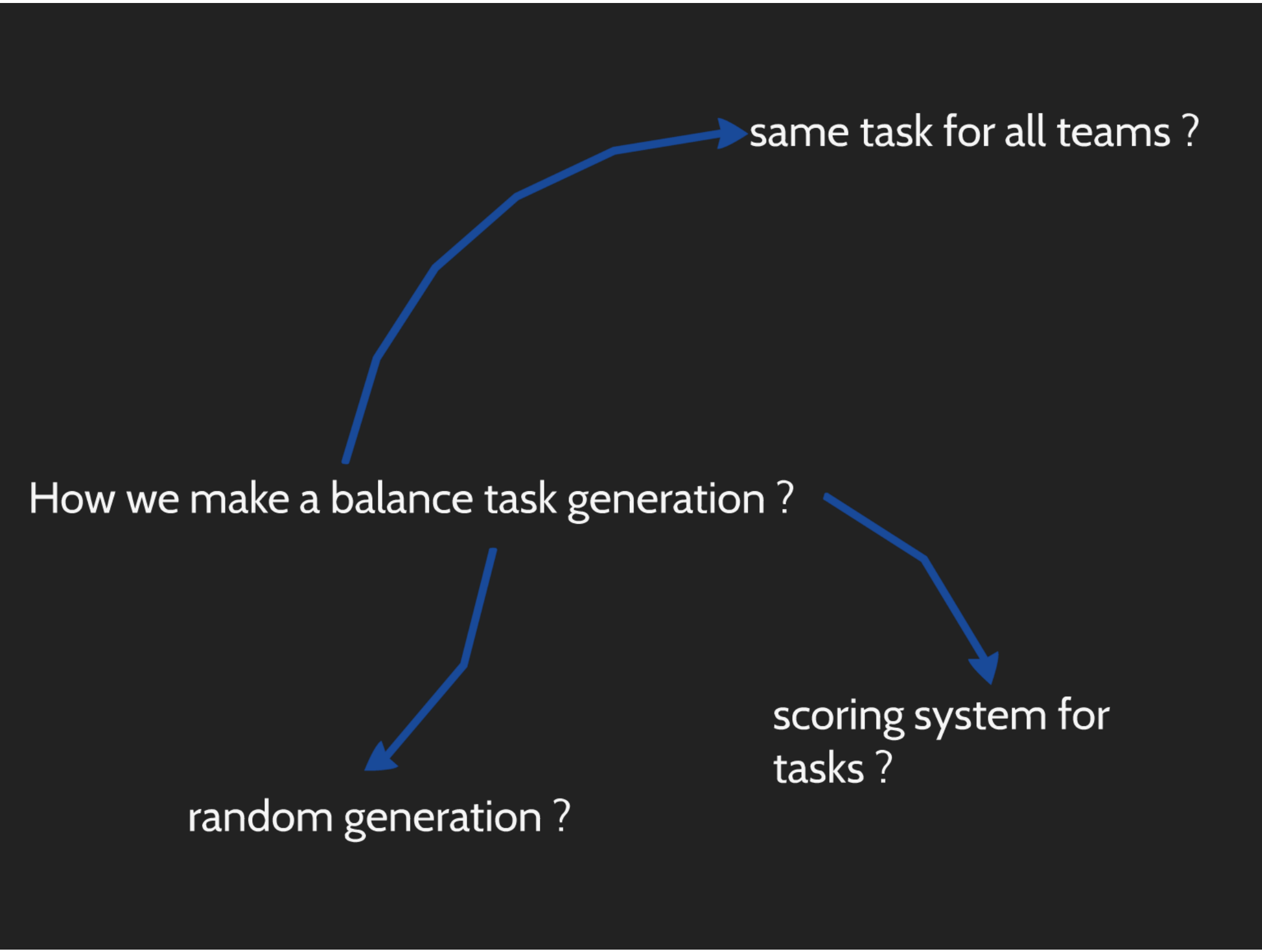


Milestones

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

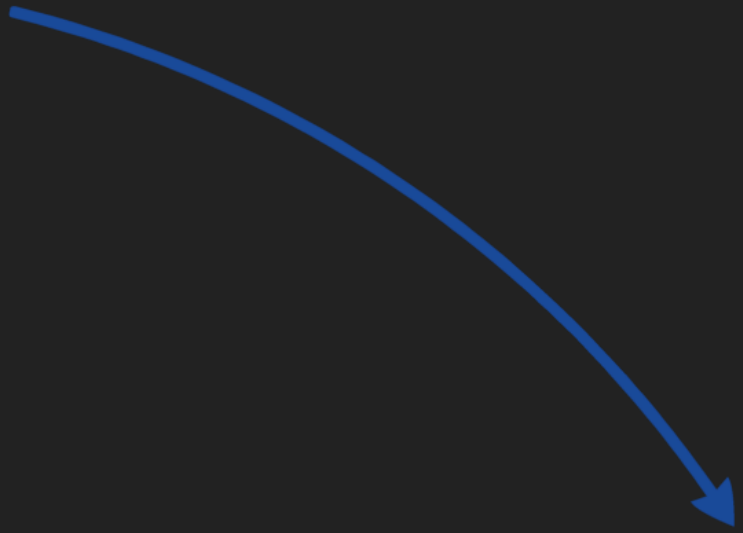
INF

robOTTO


same task for all teams ?
How we make a balance task generation ?
random generation ?
scoring system for tasks ?

Timeline

Open Questions

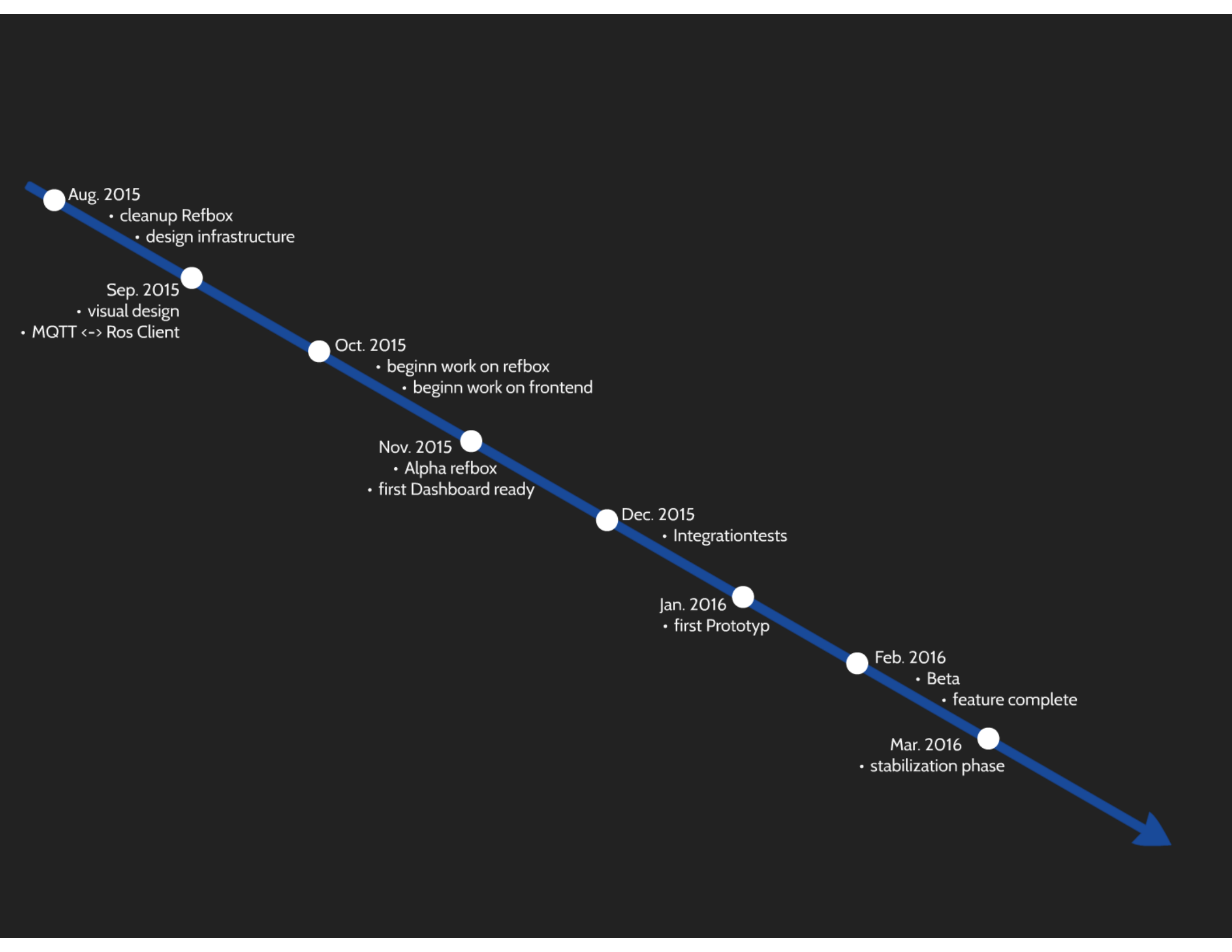


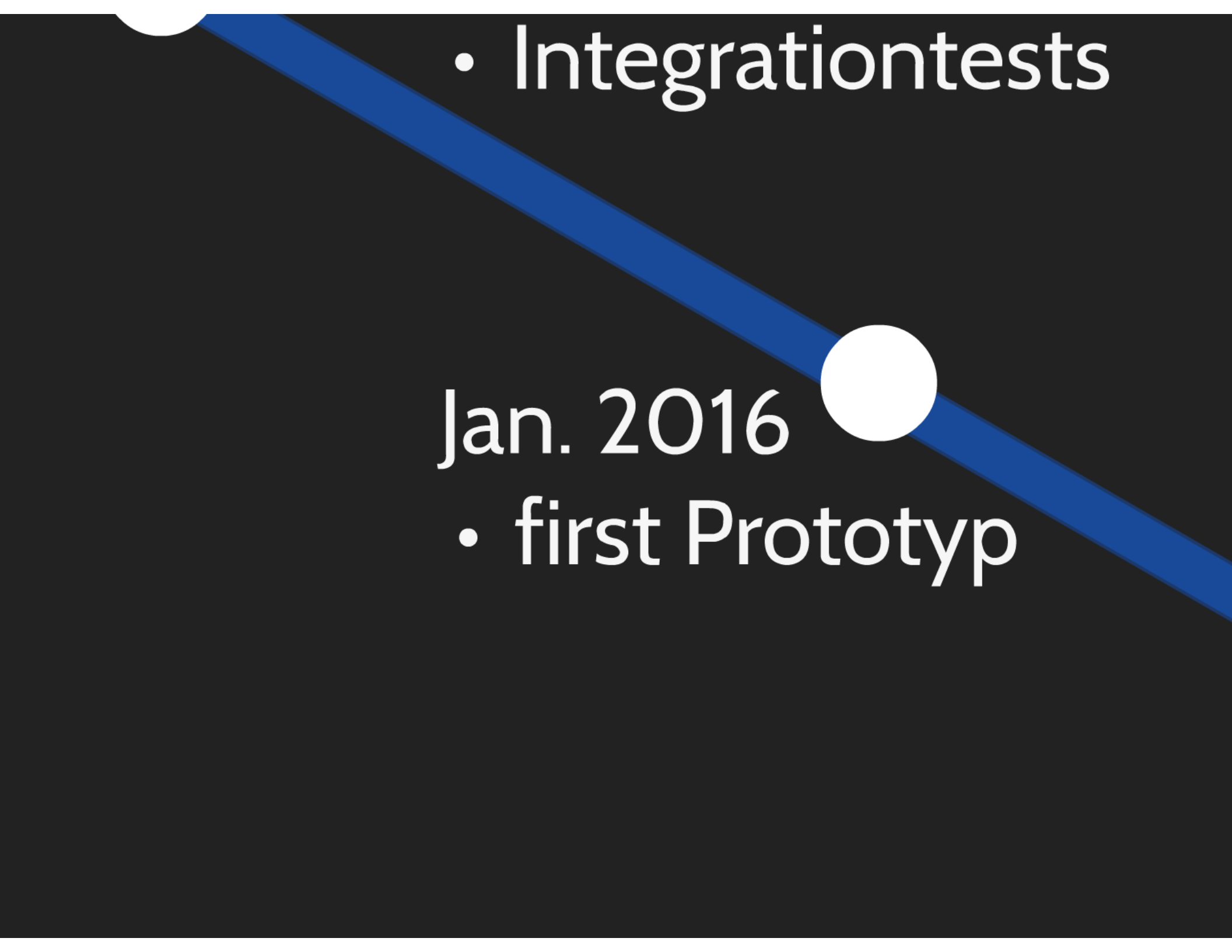
Milestones

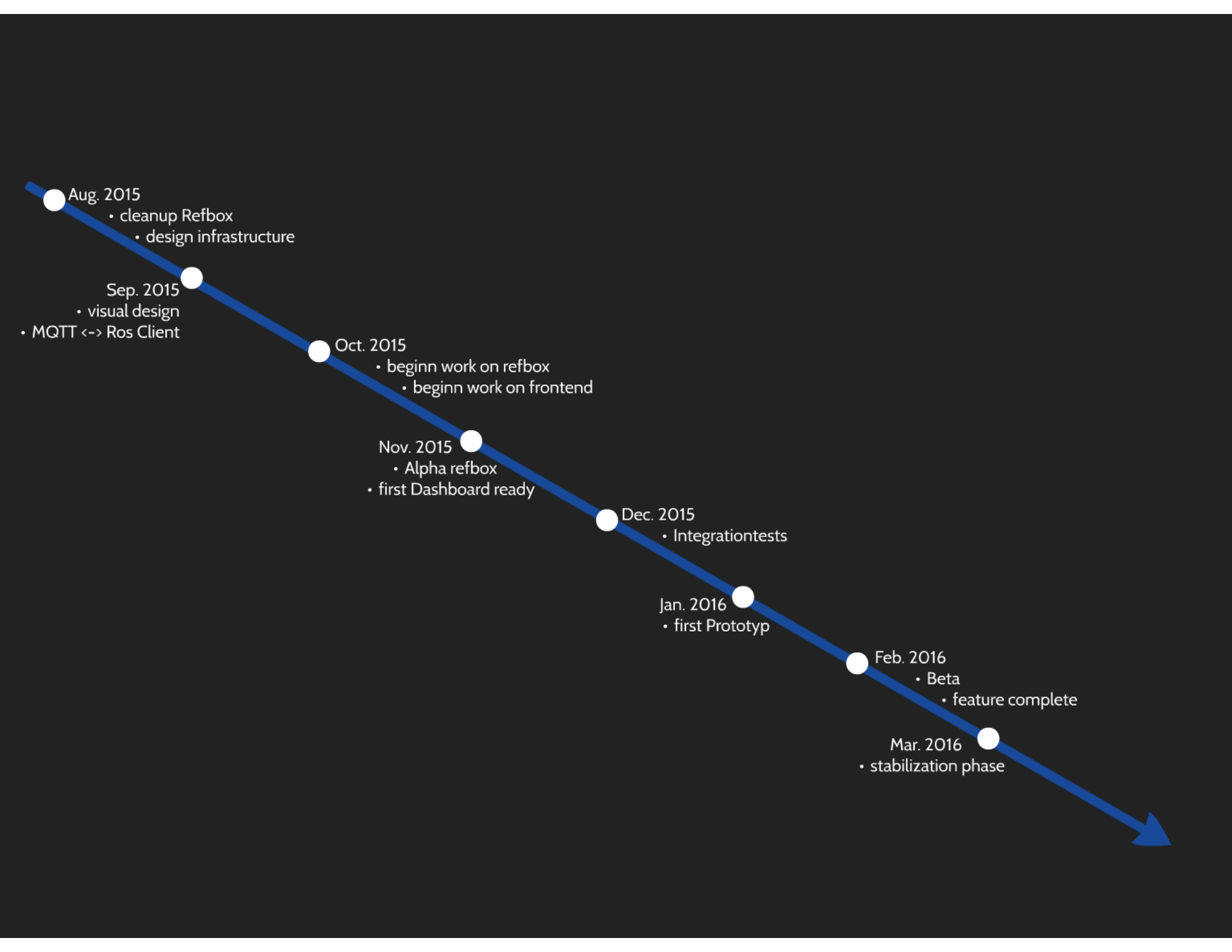
OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

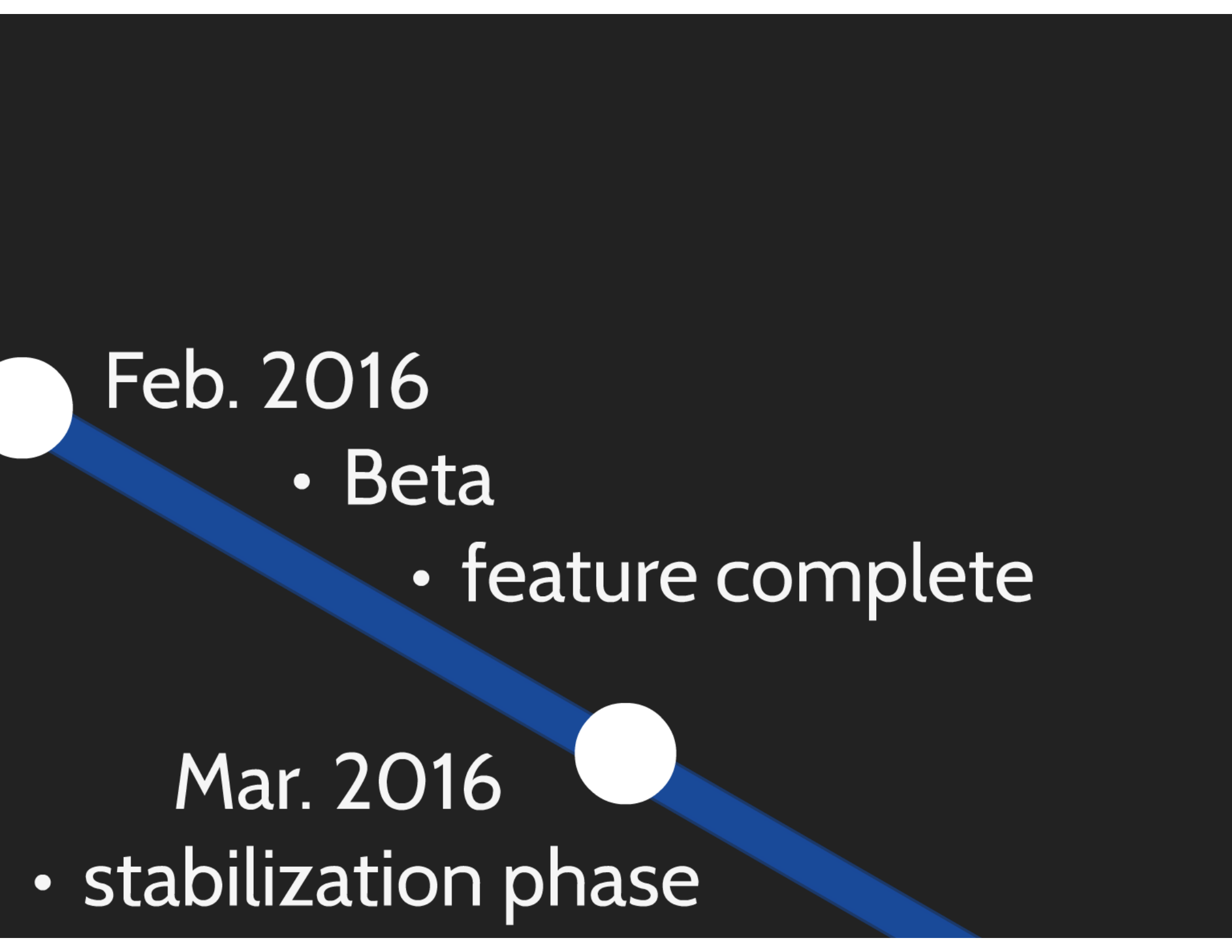
INF

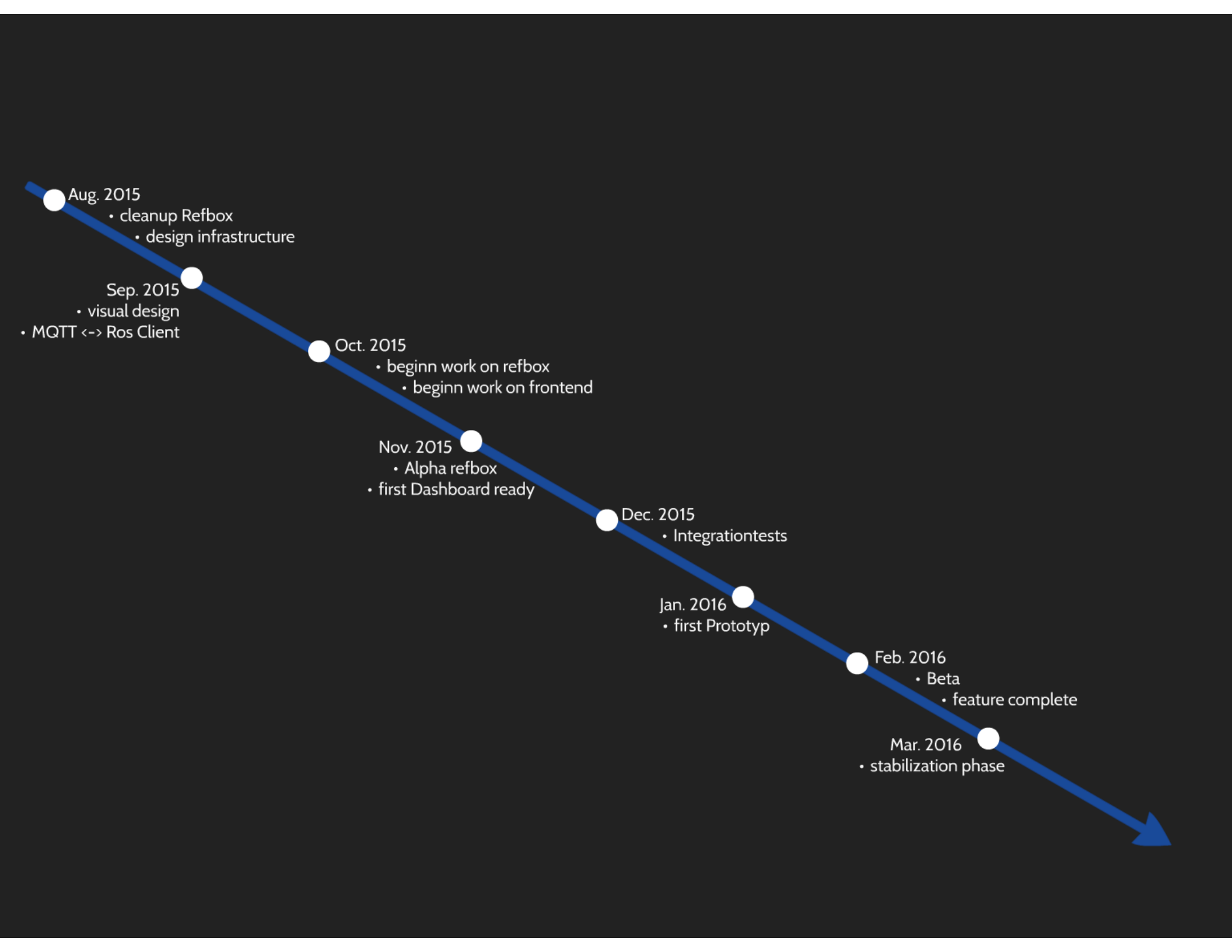
robOTTO

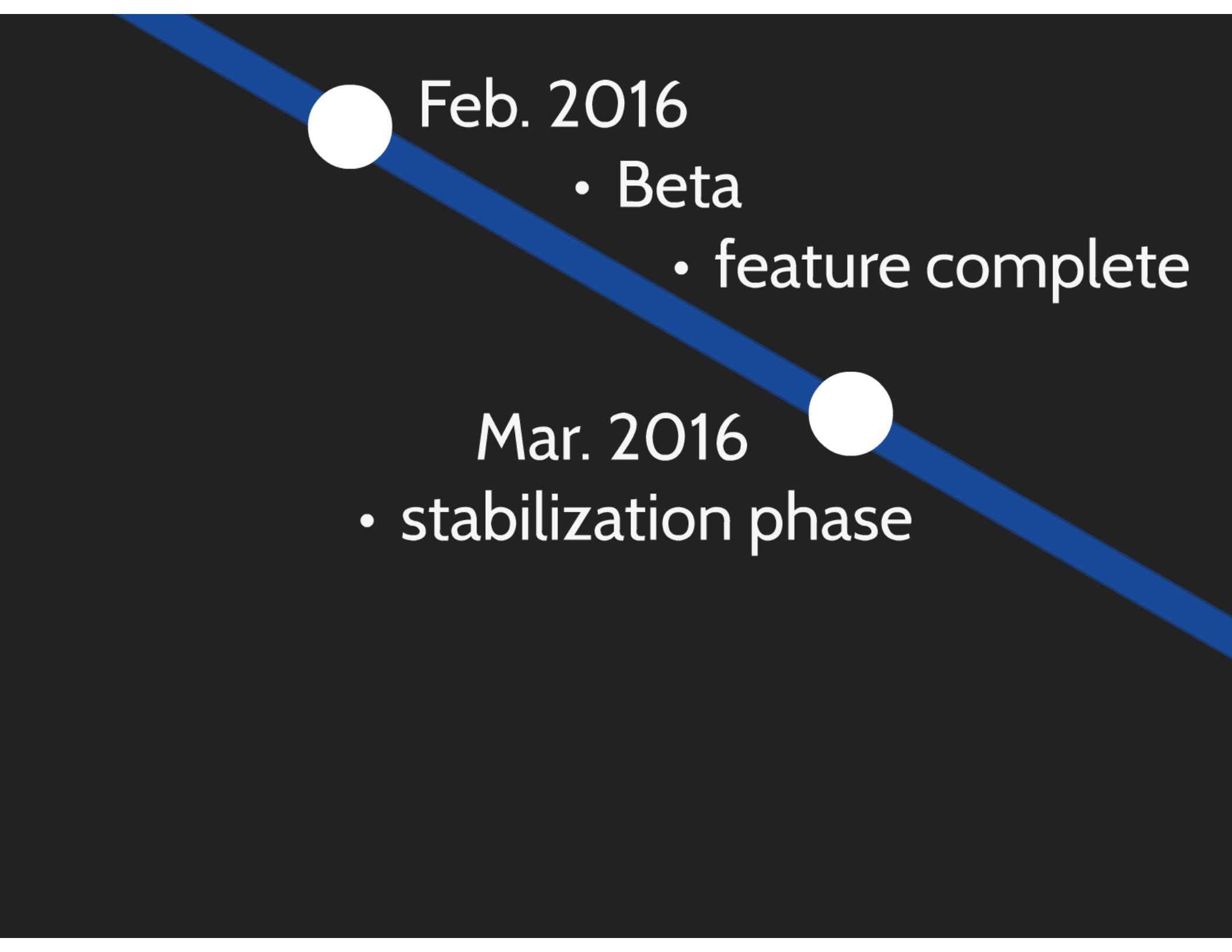

Aug. 2015
• cleanup Refbox
• design infrastructure
Sep. 2015
• visual design
• MQTT <-> Ros Client
Oct. 2015
• beginn work on refbox
• beginn work on frontend
Nov. 2015
• Alpha refbox
• first Dashboard ready
Dec. 2015
• Integrationtests
Jan. 2016
• first Prototyp
Feb. 2016
• Beta
• feature complete
Mar. 2016
• stabilization phase

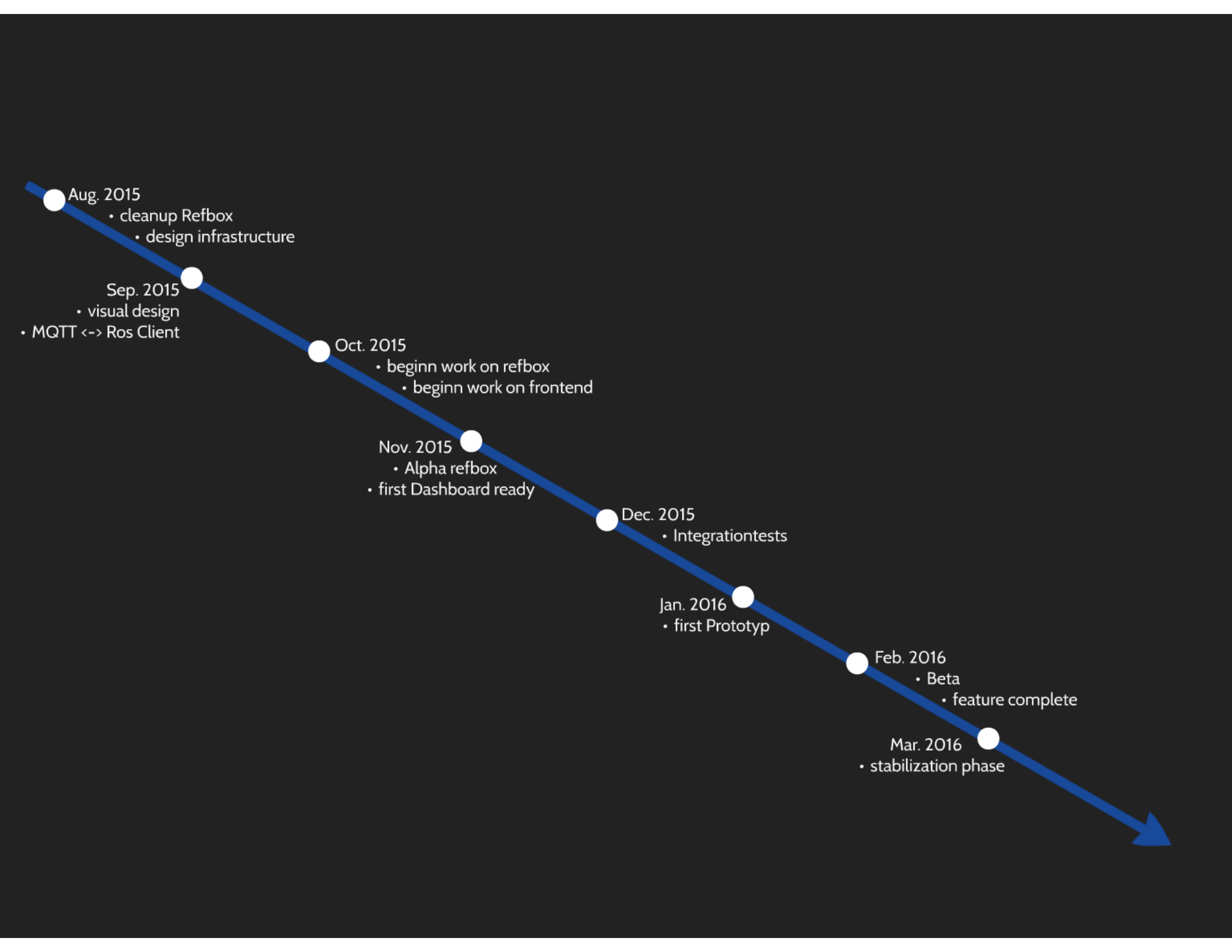

• Integrationtests
Jan. 2016
• first Prototyp


Aug. 2015
• cleanup Refbox
• design infrastructure
Sep. 2015
• visual design
• MQTT <-> Ros Client
Oct. 2015
• beginn work on refbox
• beginn work on frontend
Nov. 2015
• Alpha refbox
• first Dashboard ready
Dec. 2015
• Integrationtests
Jan. 2016
• first Prototyp
Feb. 2016
• Beta
• feature complete
Mar. 2016
• stabilization phase


Feb. 2016
• Beta
• feature complete
Mar. 2016
• stabilization phase


Aug. 2015
• cleanup Refbox
• design infrastructure
Sep. 2015
• visual design
• MQTT <-> Ros Client
Oct. 2015
• beginn work on refbox
• beginn work on frontend
Nov. 2015
• Alpha refbox
• first Dashboard ready
Dec. 2015
• Integrationtests
Jan. 2016
• first Prototyp
Feb. 2016
• Beta
• feature complete
Mar. 2016
• stabilization phase


Feb. 2016
• Beta
• feature complete
Mar. 2016
• stabilization phase


Aug. 2015
• cleanup Refbox
• design infrastructure
Sep. 2015
• visual design
• MQTT <-> Ros Client
Oct. 2015
• beginn work on refbox
• beginn work on frontend
Nov. 2015
• Alpha refbox
• first Dashboard ready
Dec. 2015
• Integrationtests
Jan. 2016
• first Prototyp
Feb. 2016
• Beta
• feature complete
Mar. 2016
• stabilization phase

Conception and Implementation of a robocup@Work refereebox

Kai Seidensticker, robOTTO

Requirements

- robcup@Work
- Referee Box Today
- Our Goals

Conception

- Overview
- Visual Design
- Feature

Timeline

- Open Questions
- Milestones

Future Work

- framework to create applications
- better design for frontend
- Morph
- eaieser deployment
  - use launchpad or reprepro

Future Work

- framework to create applications
- better design for frontend

- eaieser deployment
  - use launchpad or reprepro

Morph

- dev kit powered by bootstrap
- included MQTT support
- included all templates to create new frontend in CIS enviroment and style
- it's enabled a greate way for community work
- first prototyp: Q4 2016

Conception and Implementation of a robocup@Work refereebox

Kai Seidensticker, robOTTO

Requirements

- robcup@Work
- Referee Box Today
- Our Goals

Conception

- Overview
- Visual Design
- Feature

Timeline

- Open Questions
- Milestones

Future Work

- framework to create applications
- better design for frontend
- Morph
- eaieser deployment
  - use launchpad or reprepro