

# Cooperative programming, provisioning and control for time critical applications in Cloud

Dr. Zhiming Zhao

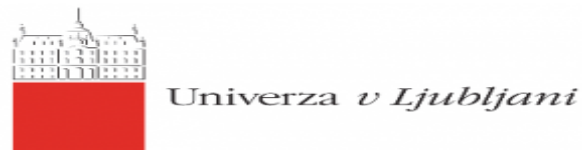
University of Amsterdam



# About the SWITCH project



- Between Feb/ 2015 – Jan/2018
- Total budget 2.92 M
- EU H2020 ICT-9
- 6 Partners
- Web: [www.switchproject.eu](http://www.switchproject.eu)

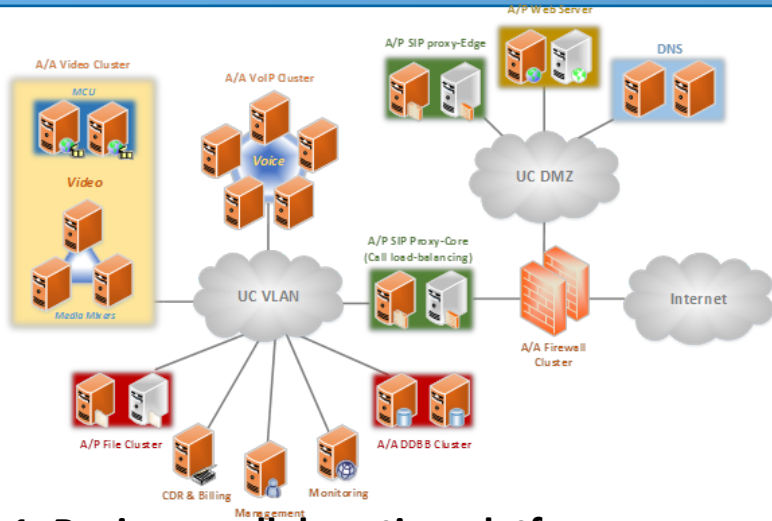




# Outline



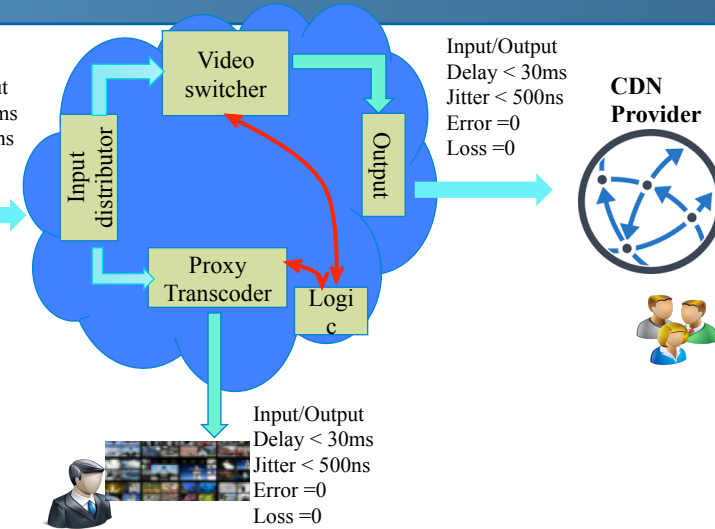
- Background: time critical applications in cloud
- Development challenges and the SWITCH approach
- Software development and exploitation
- Summary and future work



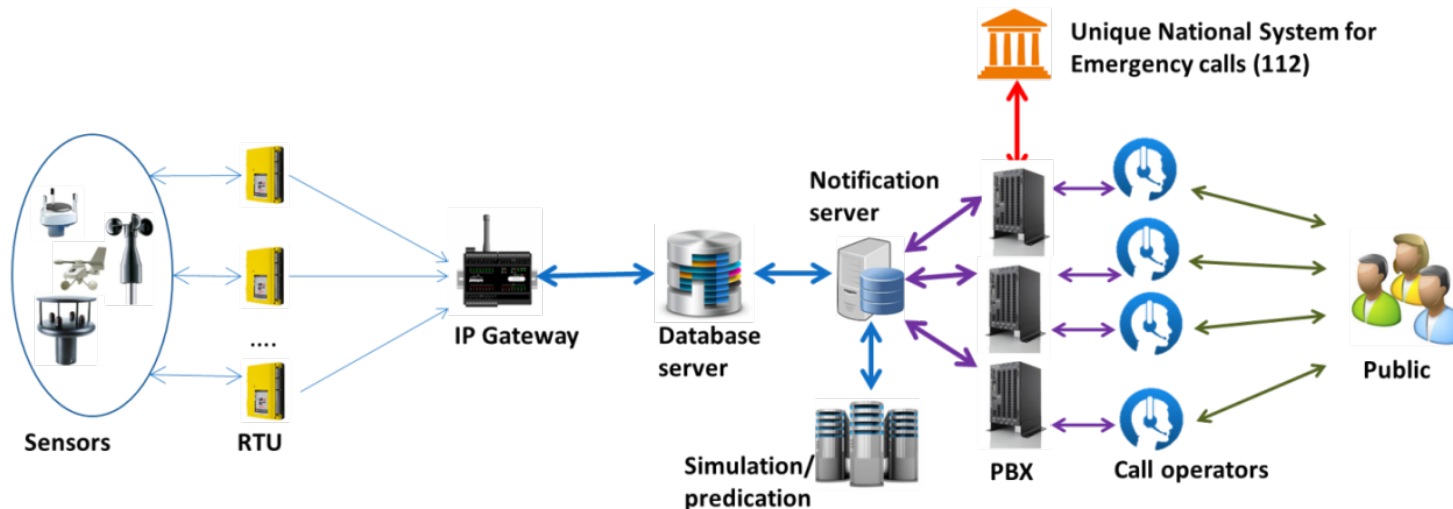
1. Business collaboration platform



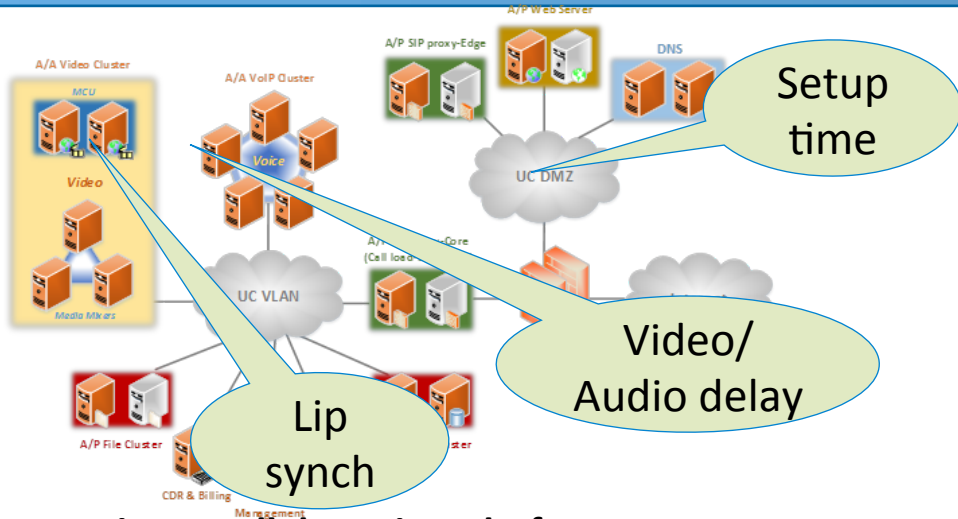
Input/Output  
Delay < 30ms  
Jitter < 500ns  
Error = 0  
Loss = 0



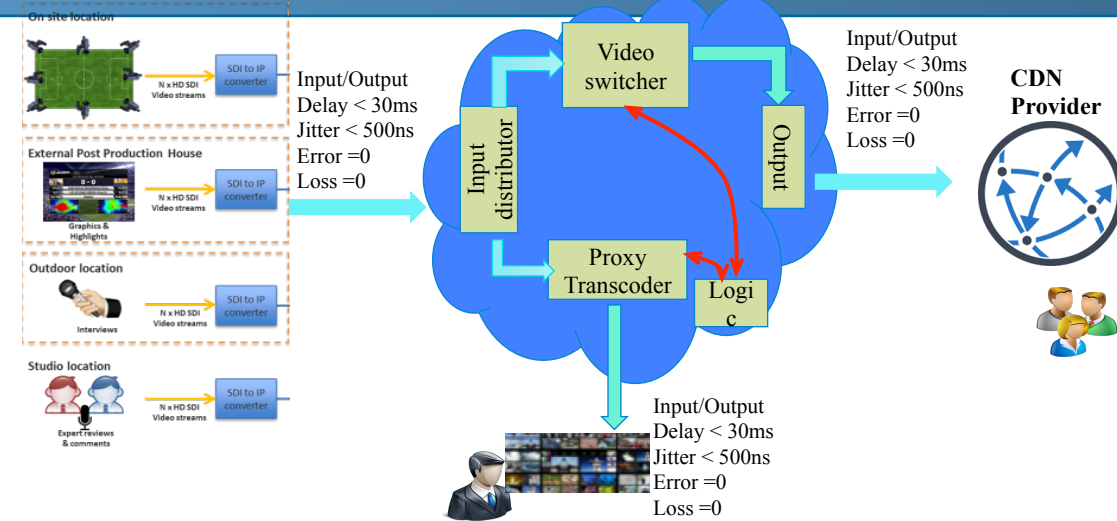
3. Directing and broadcasting live events



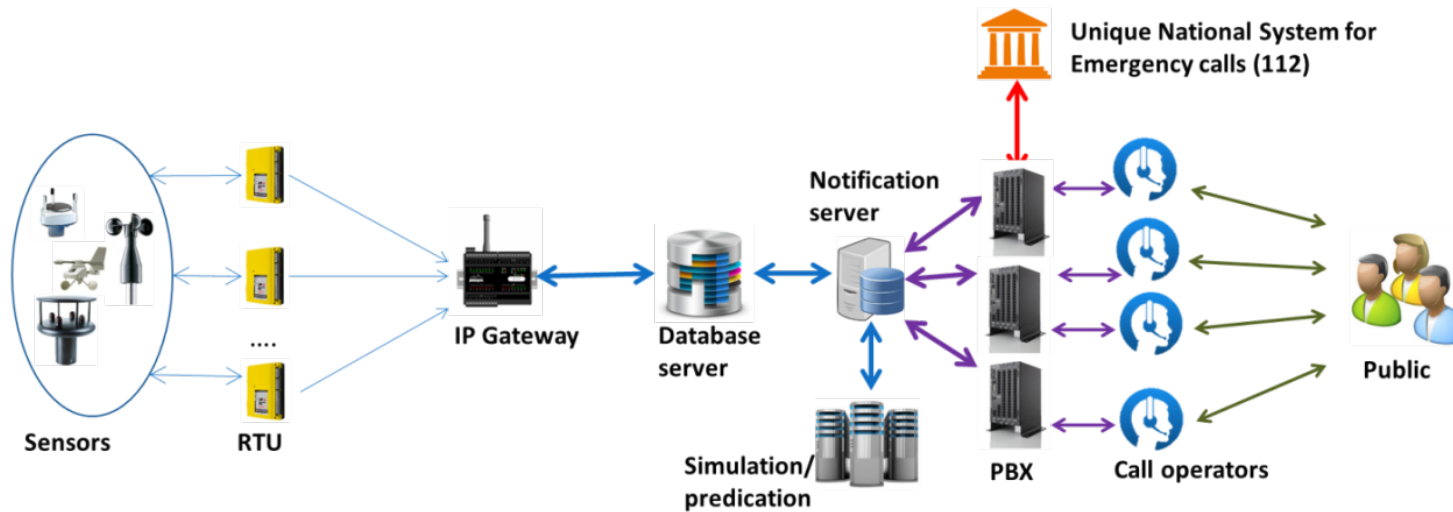
2. Disaster early warning system



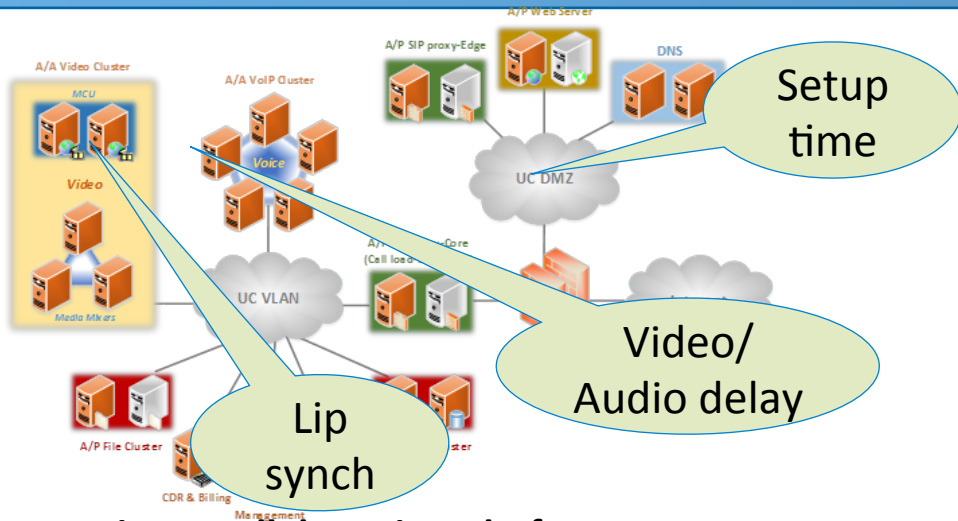
1. Business collaboration platform



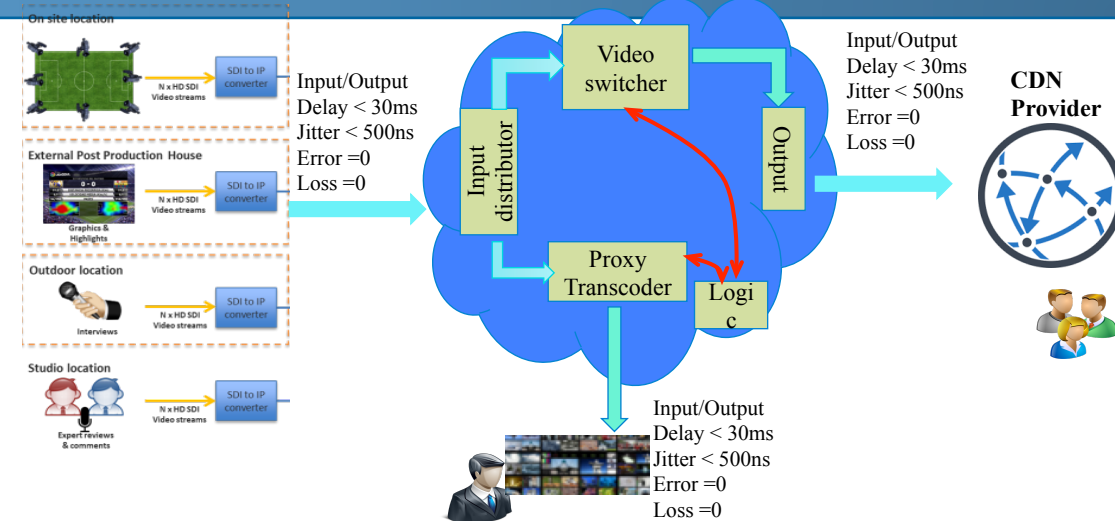
3. Directing and broadcasting live events



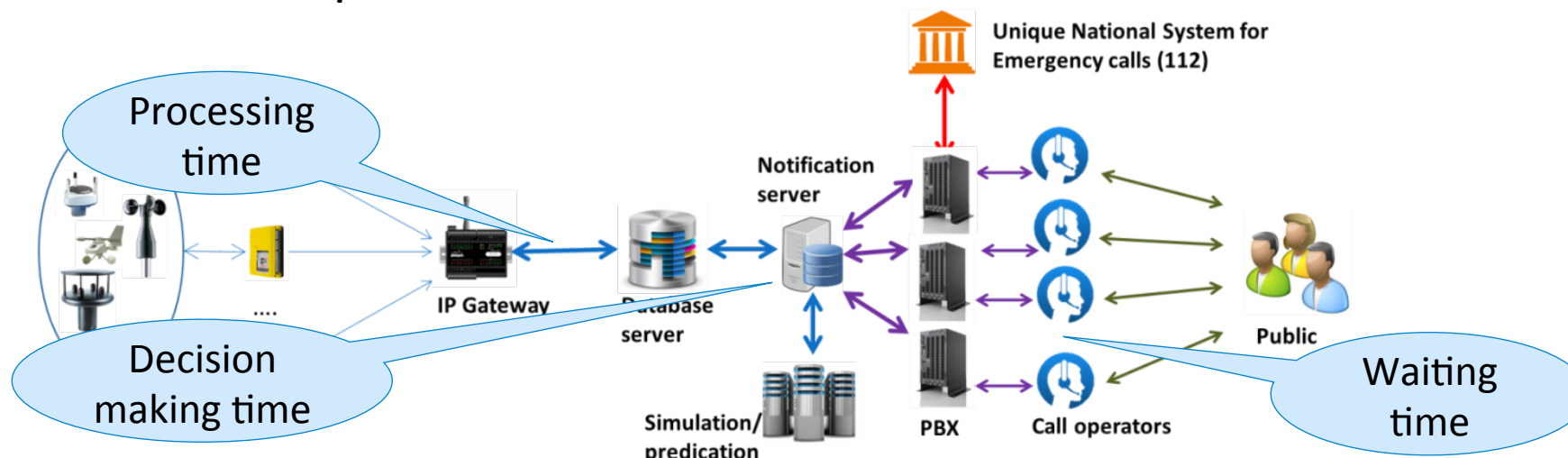
2. Disaster early warning system



1. Business collaboration platform

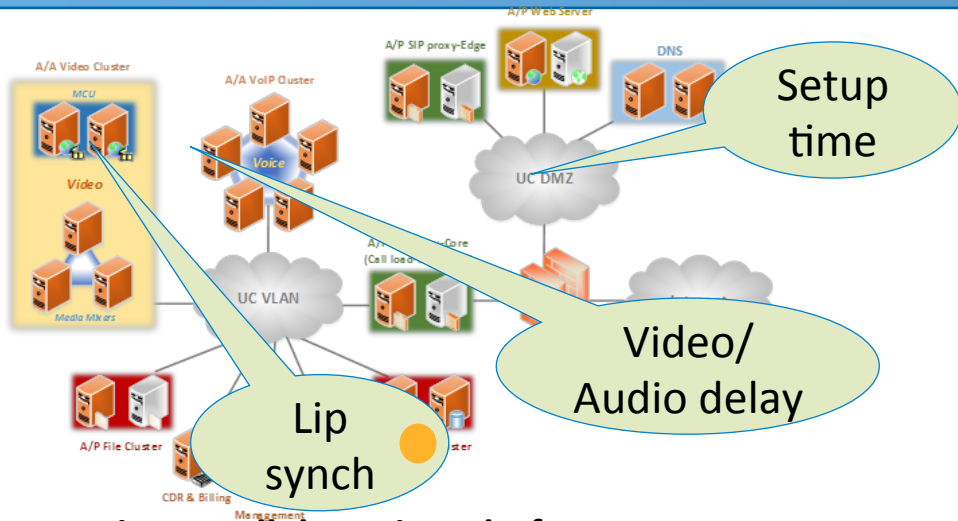


3. Directing and broadcasting live events

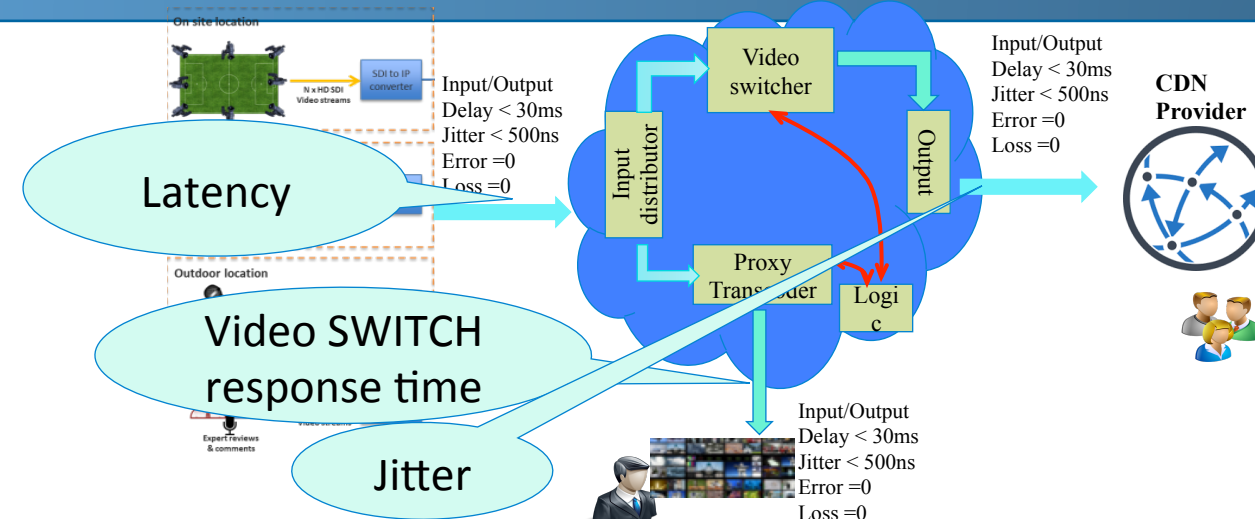


2. Disaster early warning system

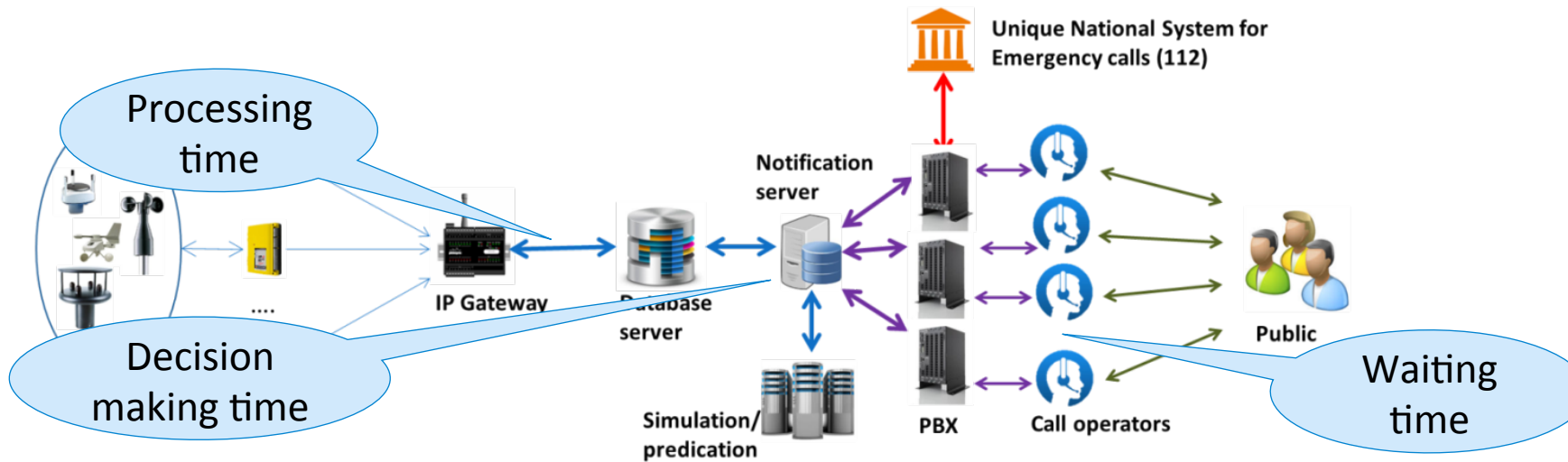




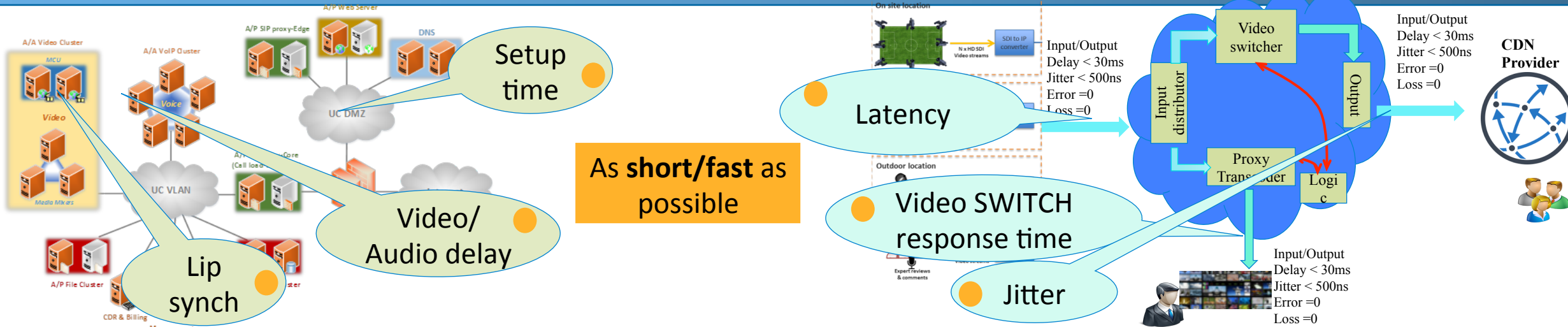
1. Business collaboration platform



3. Directing and broadcasting live events

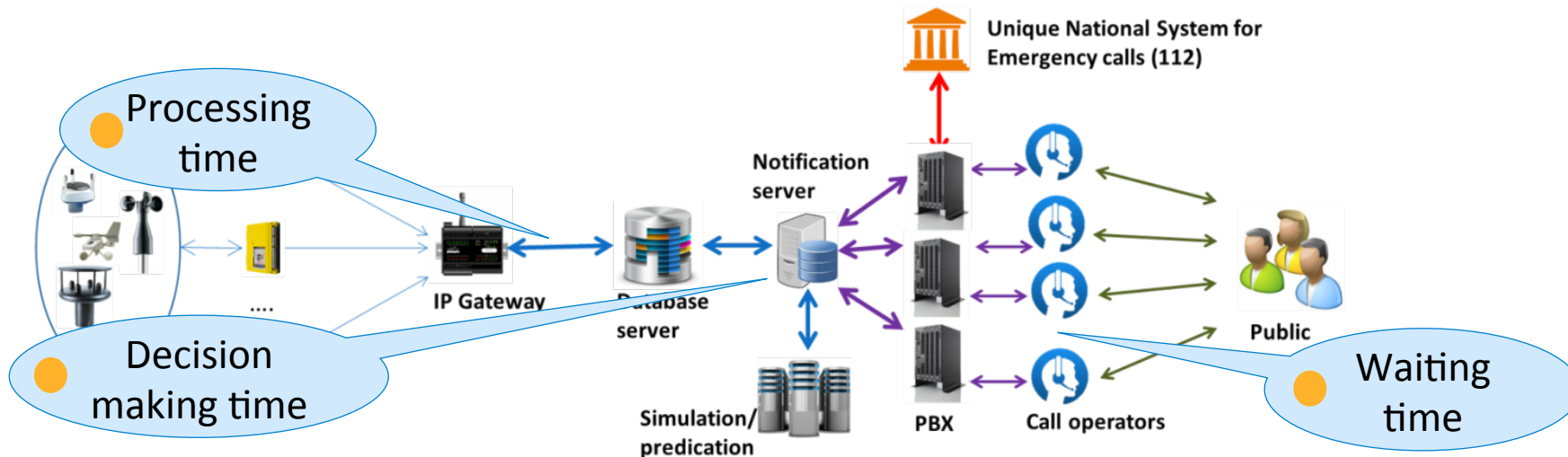


2. Disaster early warning system



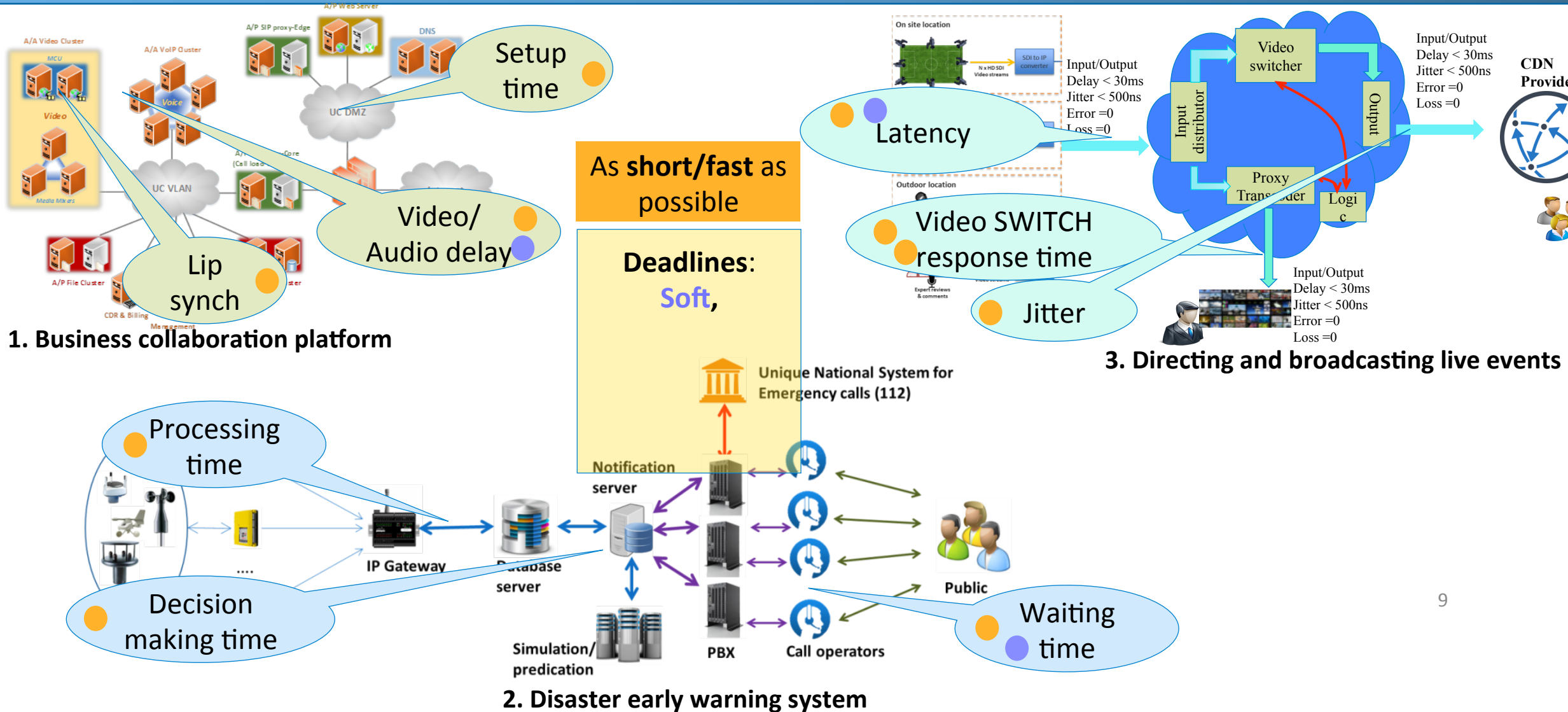
## 1. Business collaboration platform

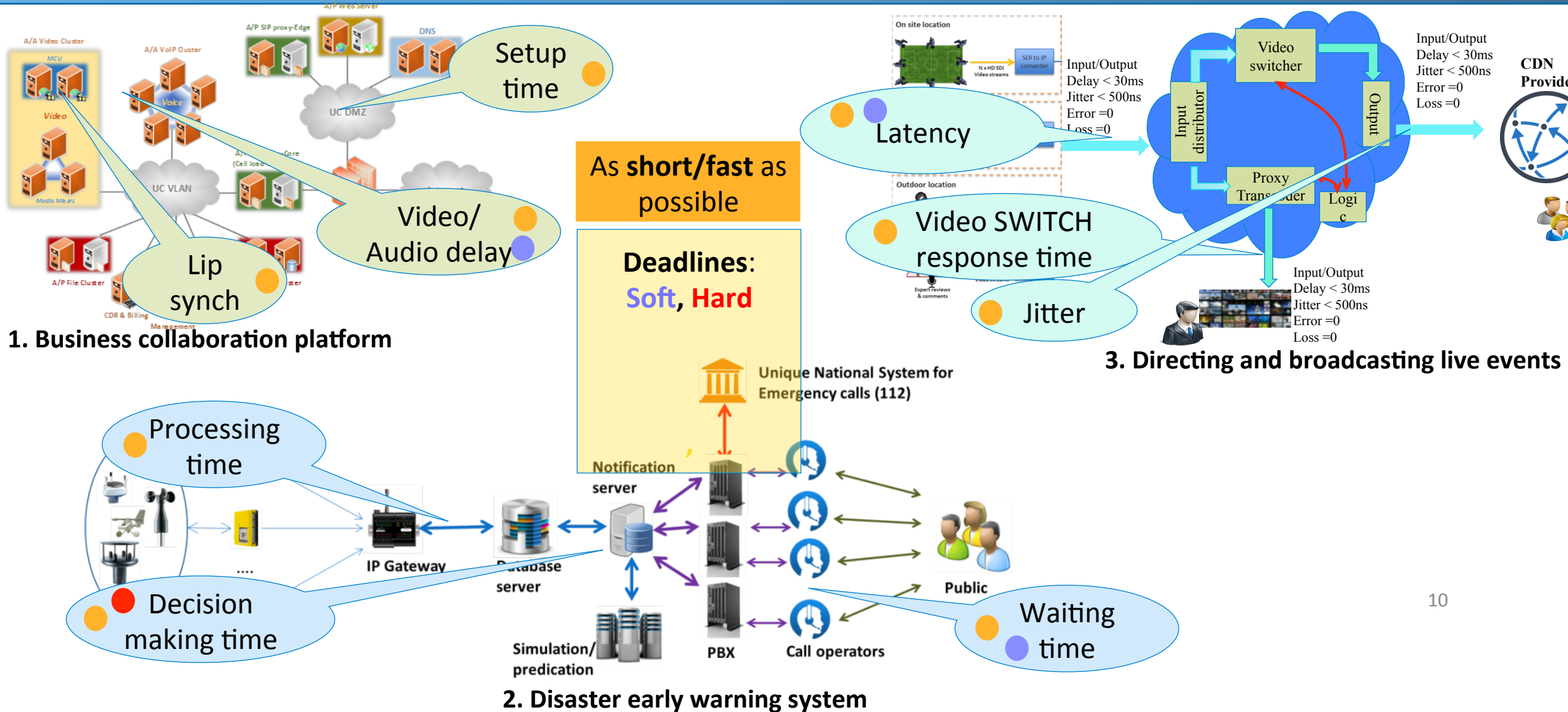
## 3. Directing and broadcasting live events

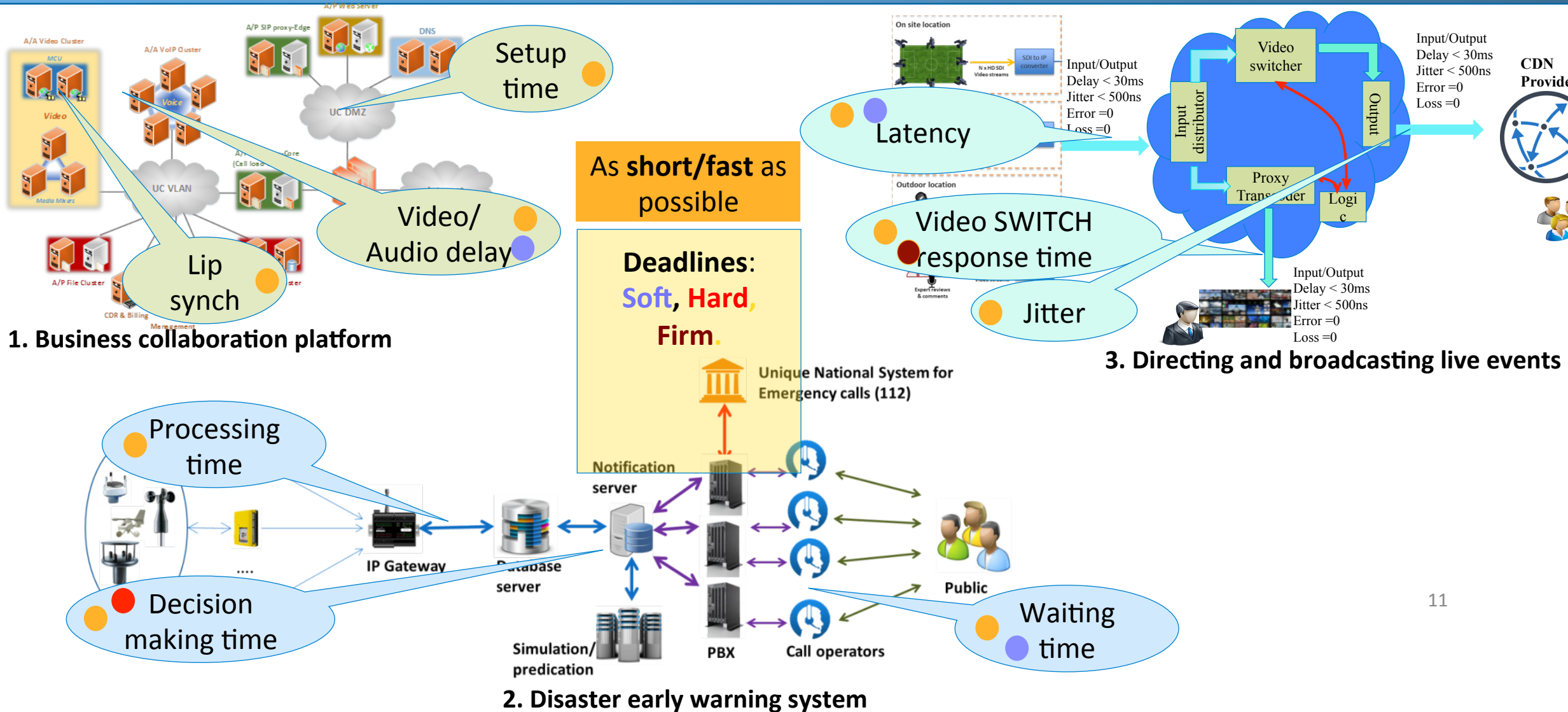


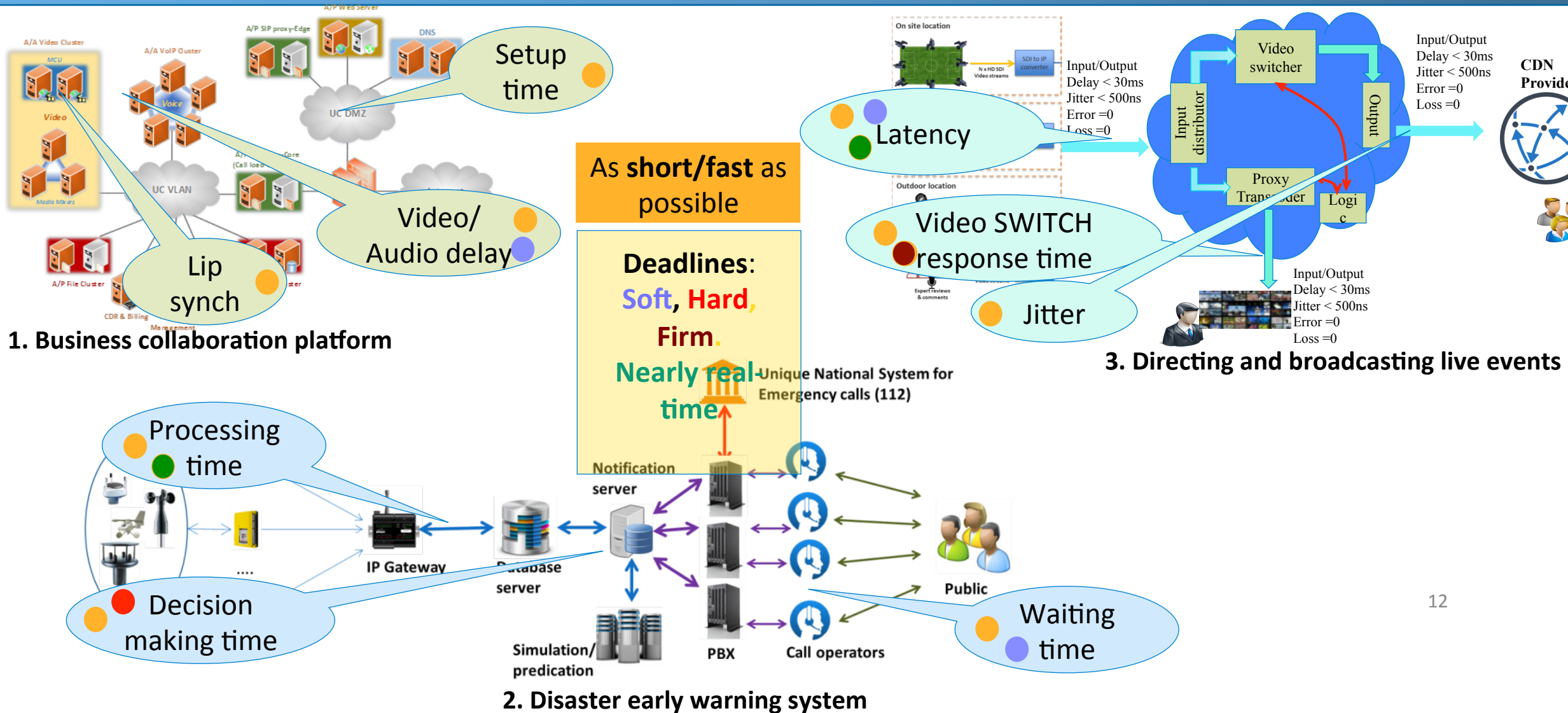
## 2. Disaster early warning system



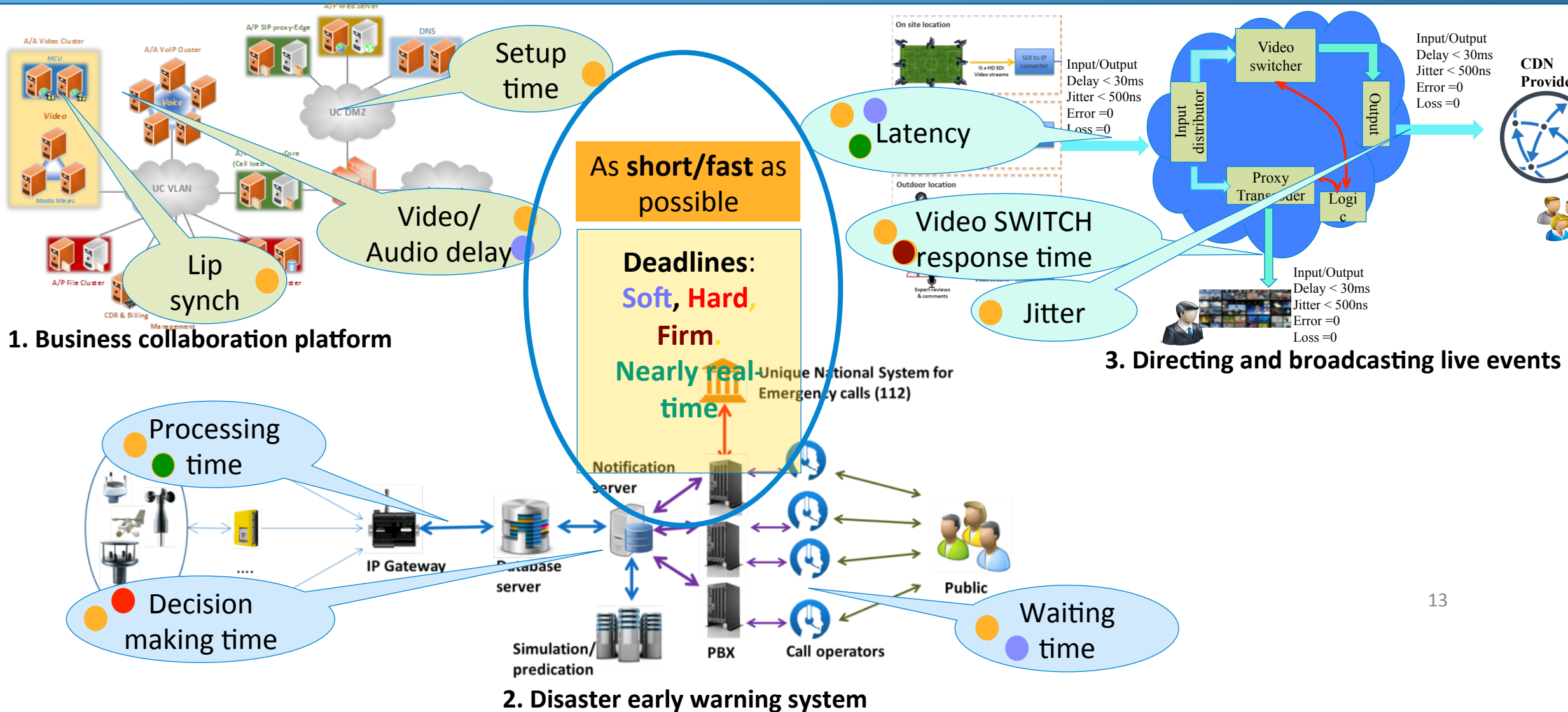


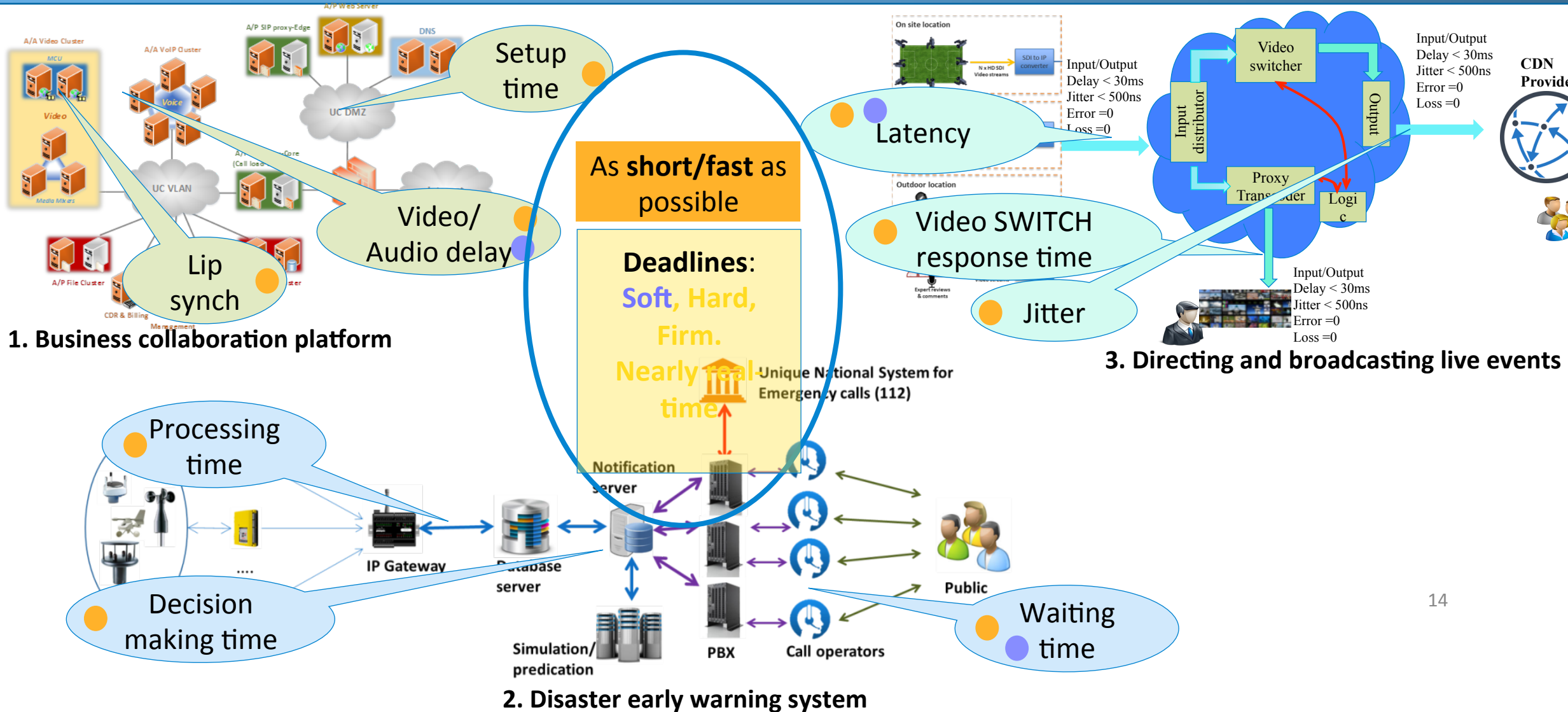






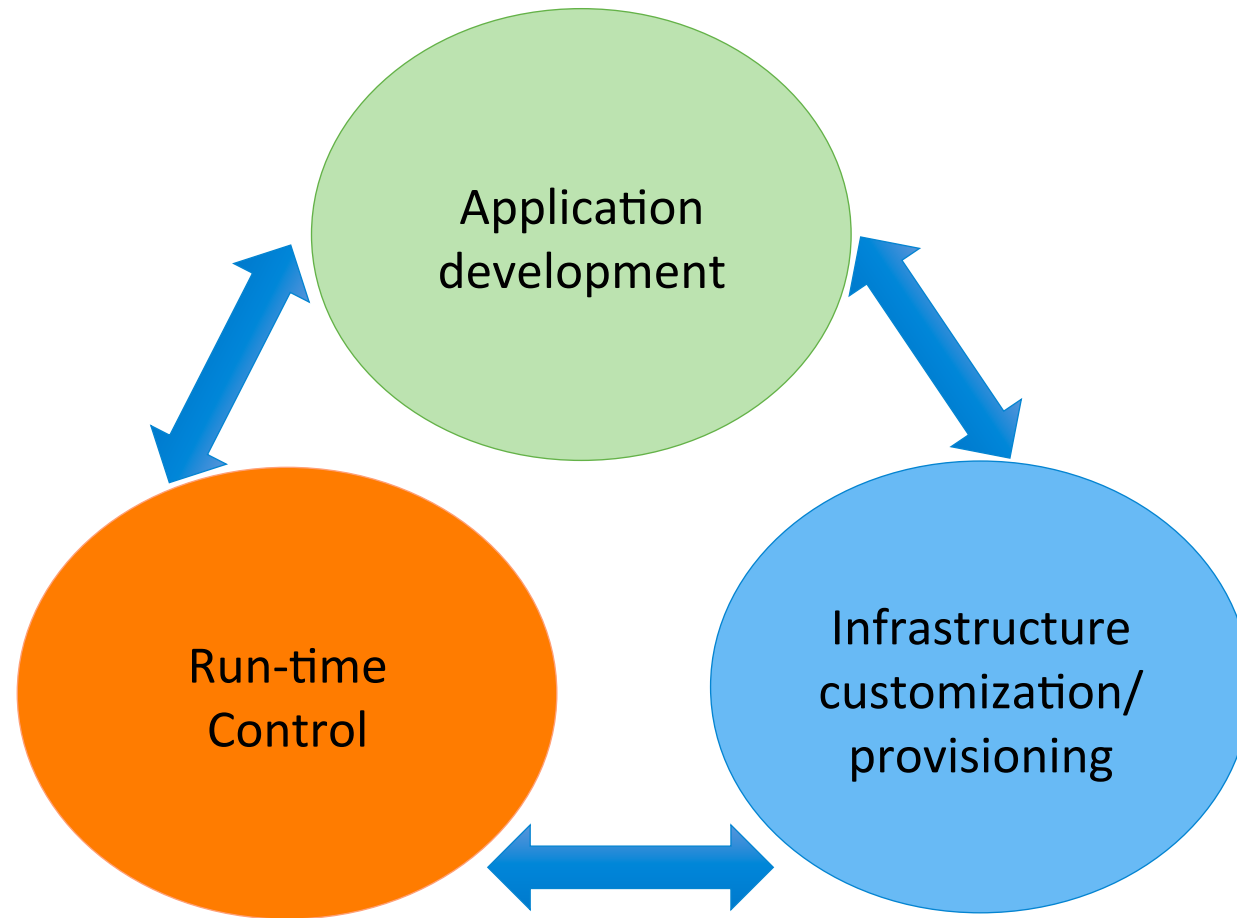








- High development and provisioning cost
  - Classical applications require special environments:
    - high performance computing and networking: dedicated hardware environment, or
    - Scheduling real-time tasks: special support from the operating systems, e.g., real-time Xen, OpenStack and real-time scheduler in Linux
  - **However**, advantages of Cloud computing, e.g., elasticity and programmability, are not taken into account
- High operation cost
  - When physical devices fail, or adaptation is required to reflect configuration changes (e.g., number of sensors and computing load) or to better maintain the QoS of the system
  - **However**, advantage of Cloud infrastructure services, e.g., performance isolation, migration, and controllability are not taken into account



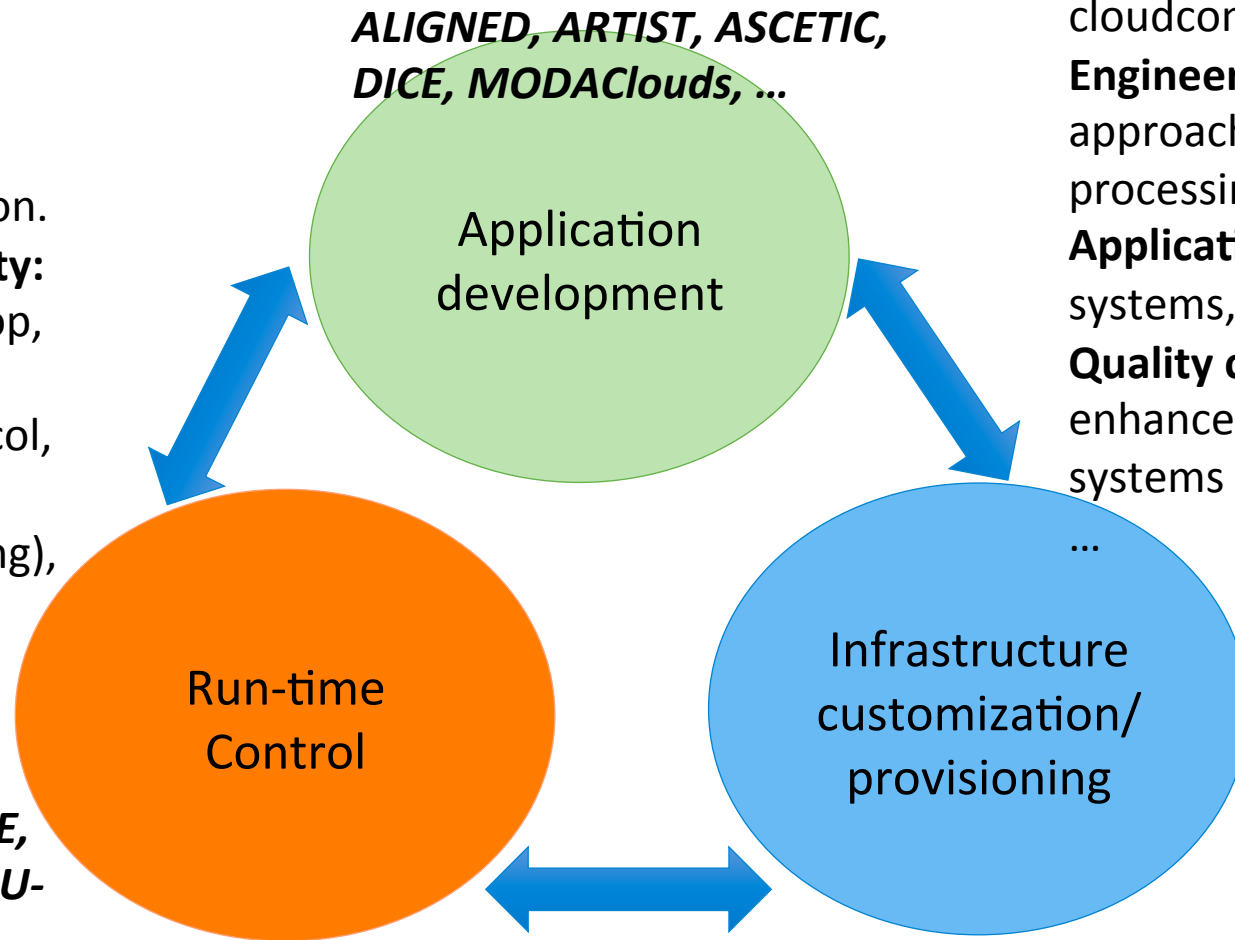


# Technology landscape and related projects



**Monitoring:** network, virtual infrastructure, cloud, application.  
**Programmability/controllability:** visualization, human-in-the-loop, system model based,  
**Control level:** network (protocol, topology etc.), VM and container (elasticity, auto-scaling), application (steering, )  
 ...

*CloudLightening, DICE, HyVar, Mikelangelo, U-QASAR, mOSAIC, etc.*



*ALIGNED, ARTIST, ASCETIC, DICE, MODAClouds, ...*

Application development

Run-time Control

Infrastructure customization/provisioning

*Paasage, Cyclone, ENTICE, SLAReady, IOStack, NetIDE...*

**Patterns:** cloud design patterns, [cloudcomputingpatterns.org](http://cloudcomputingpatterns.org)

**Engineering approach:** Model driven approach, DevOps, Open Distributed processing, Micro services

**Application execution:** workflow systems, mapReduced, message passing

**Quality critical aspects:** QoS/QoE enhance optimization, real-time systems

**Virtual infrastructure:** IaaS, VM, Container, SDN, NVFI,

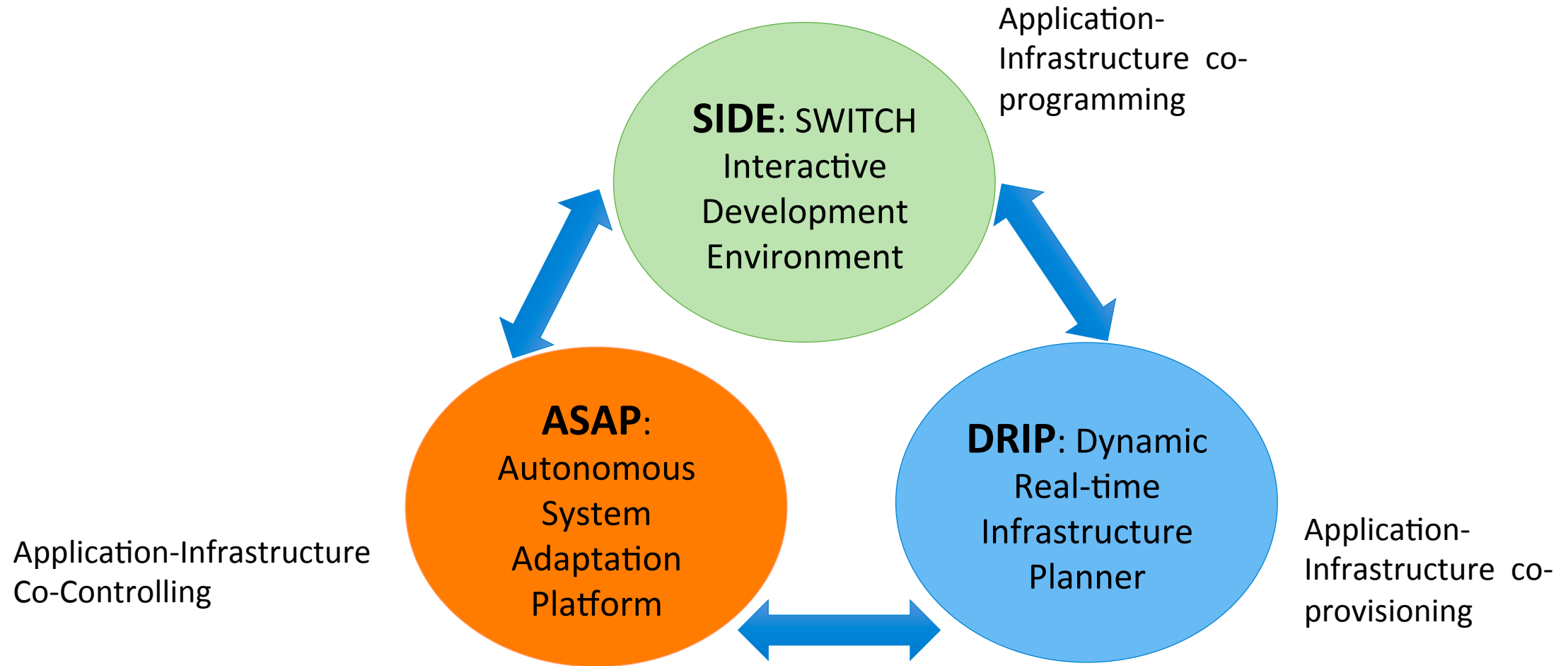
**Cloud management:** OpenStack, CloudStack, OpenNebula etc.

**Quality critical aspects:** Openflow + QoS, traffic engineering, real time virtualization

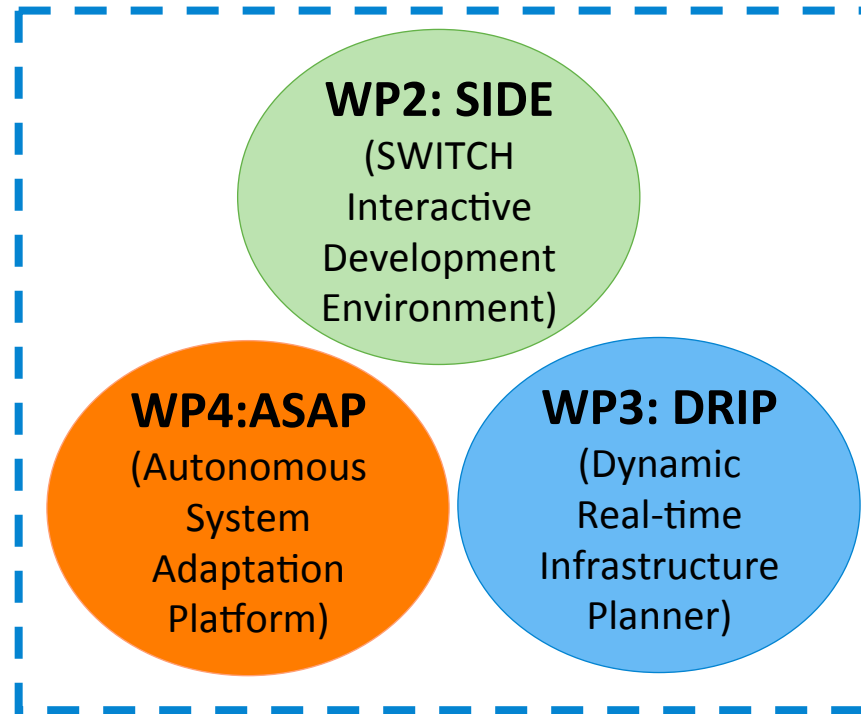
**Provisioning:** OCCI, ...

# The basic idea of SWITCH

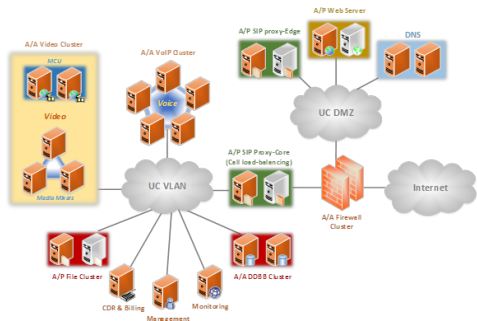
- **Addressing the entire lifecycle** of time critical cloud applications by a new software workbench (including middleware and front end tools)
  - **Improve the development and execution model** of time critical applications in programmable, elastic and virtualized environments.
  - **Improve the productivity of application development** by introducing the SWITCH Interactive Development Environment (SIDE)
  - **Improve the deployment and provisioning efficiency** by introducing the Distributed Real-time Infrastructure Planner (DRIP)
  - **Improve the QoS/QoE control efficiency** using the Autonomous System Adaptation Platform (ASAP)
  - **Promote and exploit the SWITCH methods and software** by actively combining technology development and interacting with the pilot use case providers.



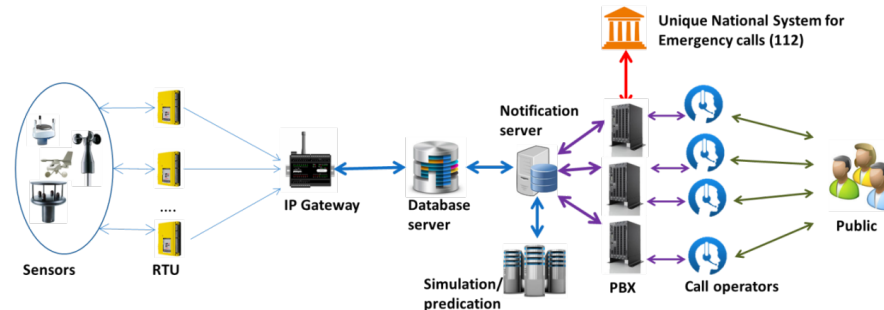
# Project structure



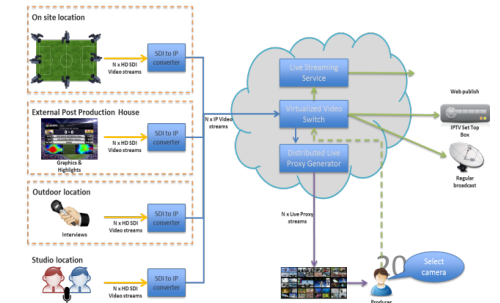
## WP 5: software test and pilot applications



1. Real-time business communication platform (Wellness Telecom, Spain)



2. Elastic disaster early warning system (BEIA, Romania)



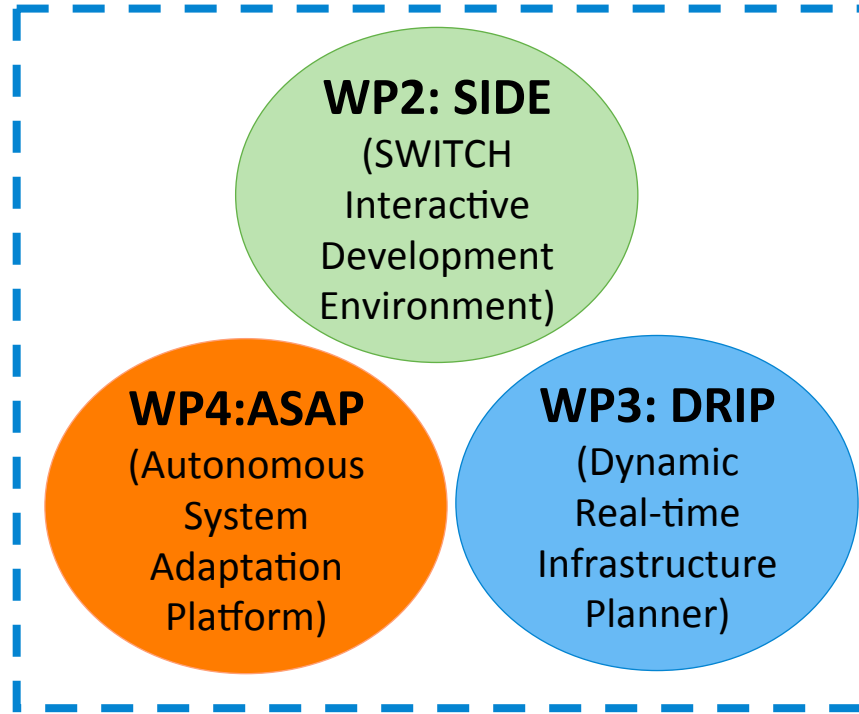
3. Cloud studio for directing and broadcasting live events (MOG, Portugal)



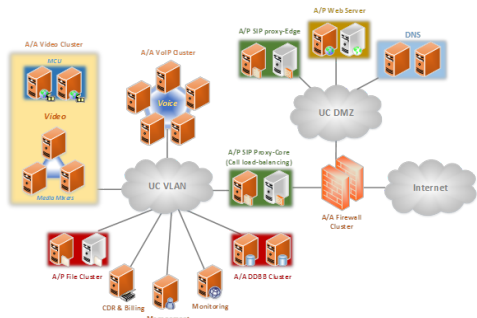
# The Project structure

Requirements for applications  
and software quality/license

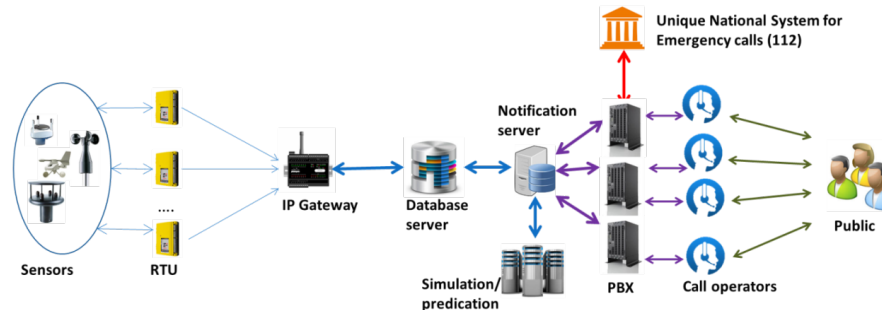
Validation of the workbench



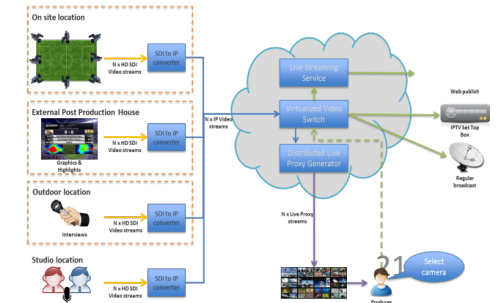
## WP 5: software test and pilot applications



1. Real-time business communication platform (Wellness Telecom, Spain)



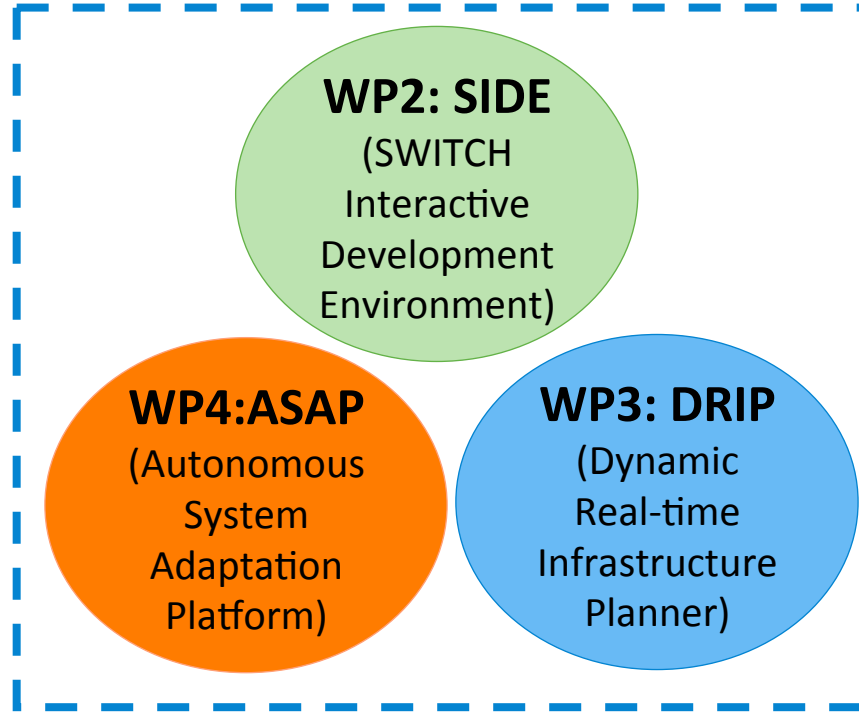
2. Elastic disaster early warning system (BEIA, Romania)



3. Cloud studio for directing and broadcasting live events (MOG, Portugal)

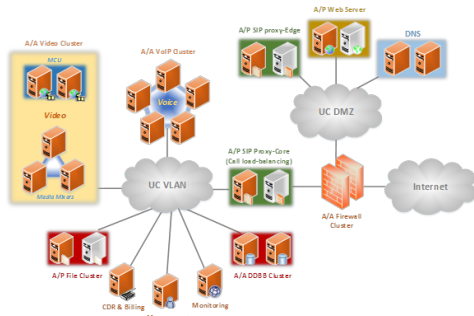
# The Project structure

Requirements for applications and software quality/license  
Validation of the workbench

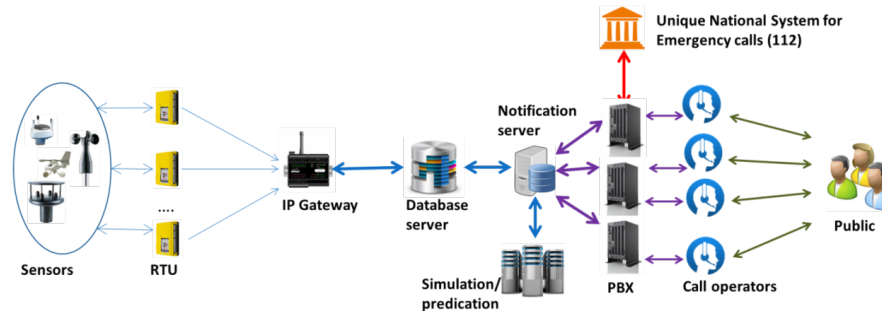


Exploitation and sustainability  
Innovation and business cases

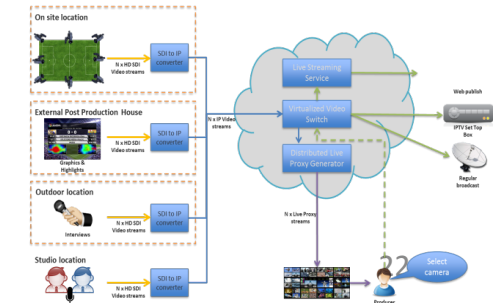
## WP 5: software test and pilot applications



1. Real-time business communication platform (Wellness Telecom, Spain)



2. Elastic disaster early warning system (BEIA, Romania)



3. Cloud studio for directing and broadcasting live events (MOG, Portugal)

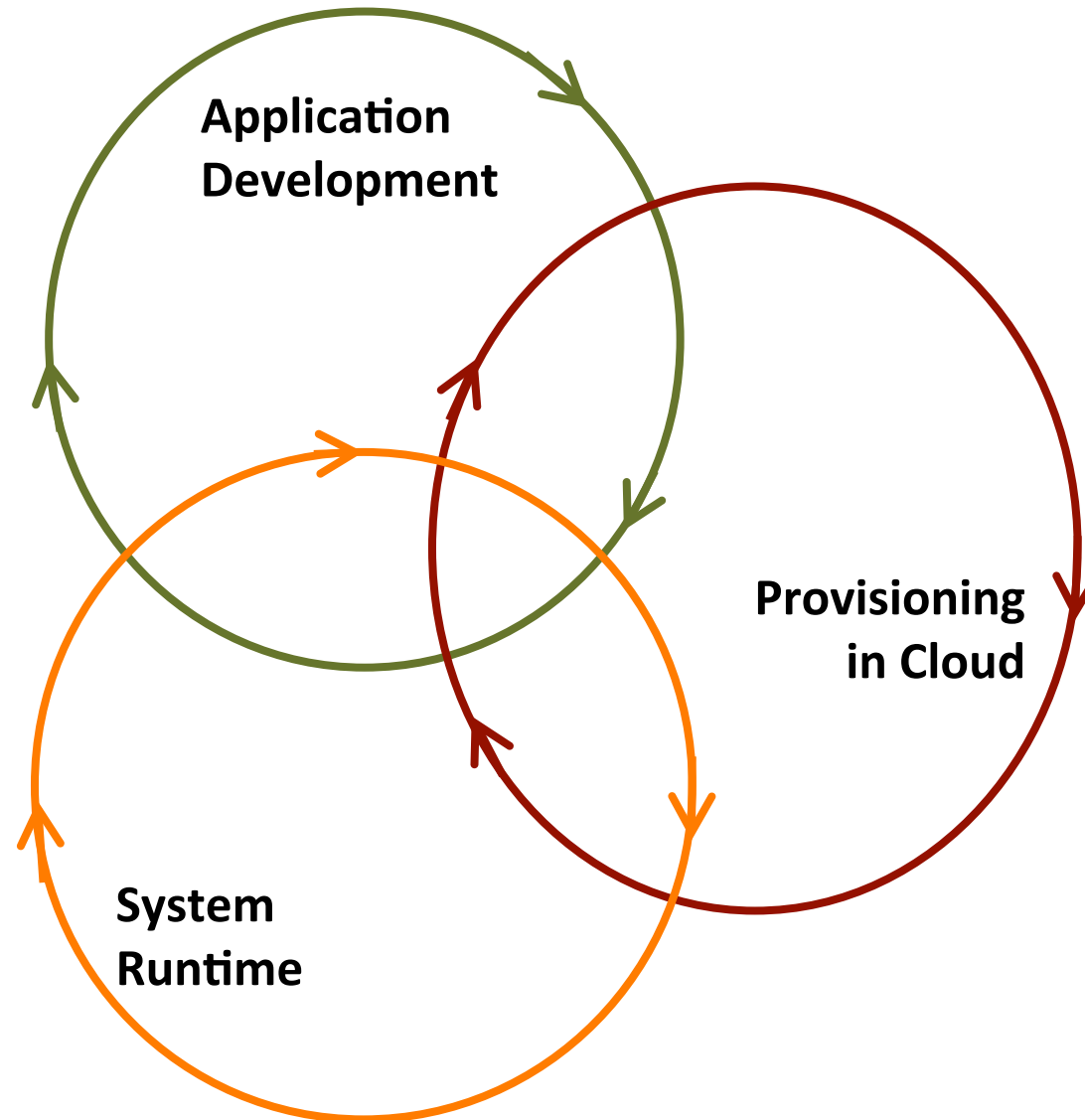


# SIDE(Co-programming): programming application-infrastructure

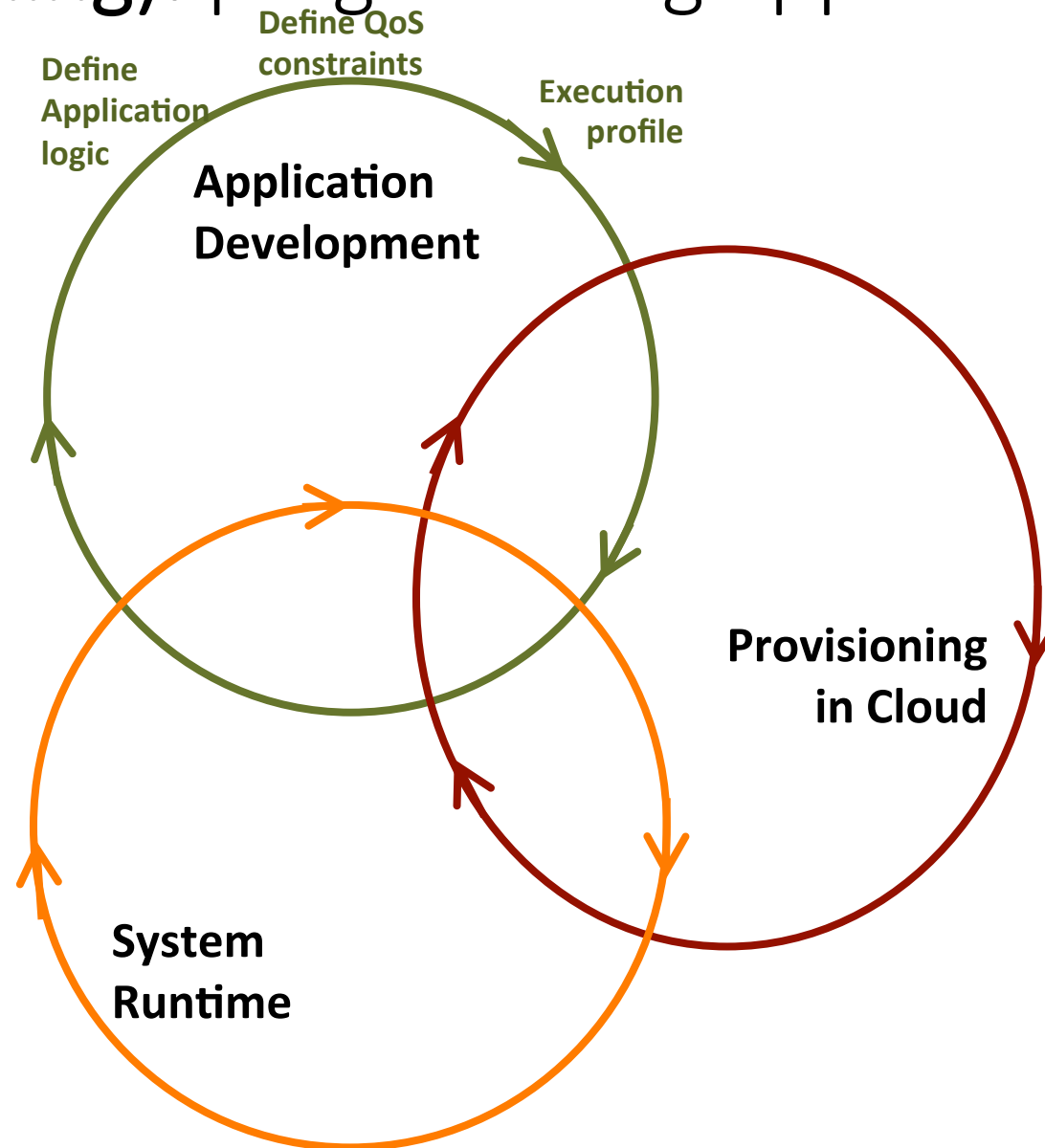


- **Scenario 1:** given a legacy application (application workflow, QoS constraints)
  - program a suitable virtual infrastructure (VM, Containers, topologies)
- **Scenario 2:** build an application with existing components (containers)
  - Program application logic (workflows) with guidance of virtual infrastructure constraints

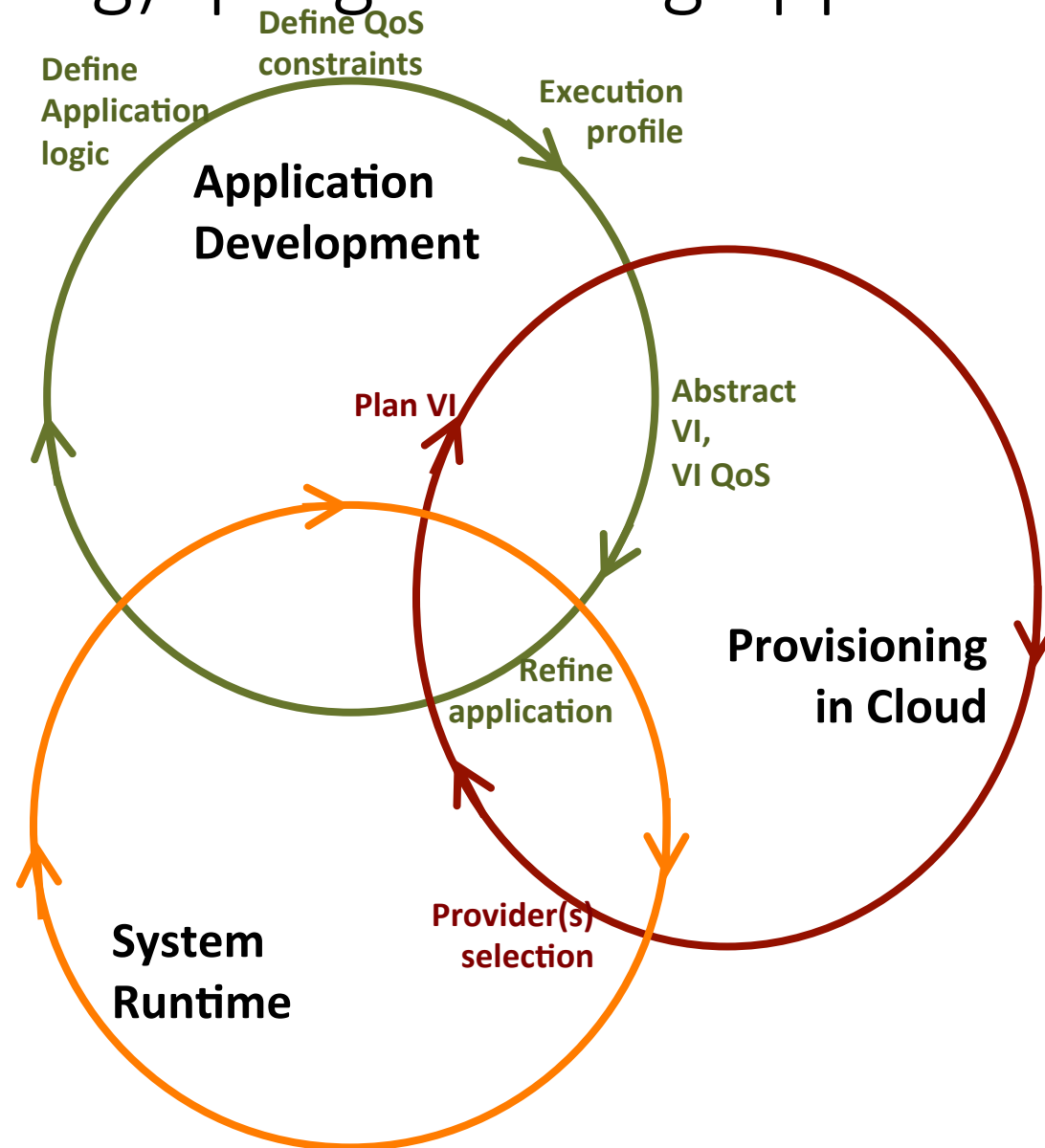
# SIDE(Co-programming): programming application-infrastructure



# SIDE(Co-programming): programming application-infrastructure

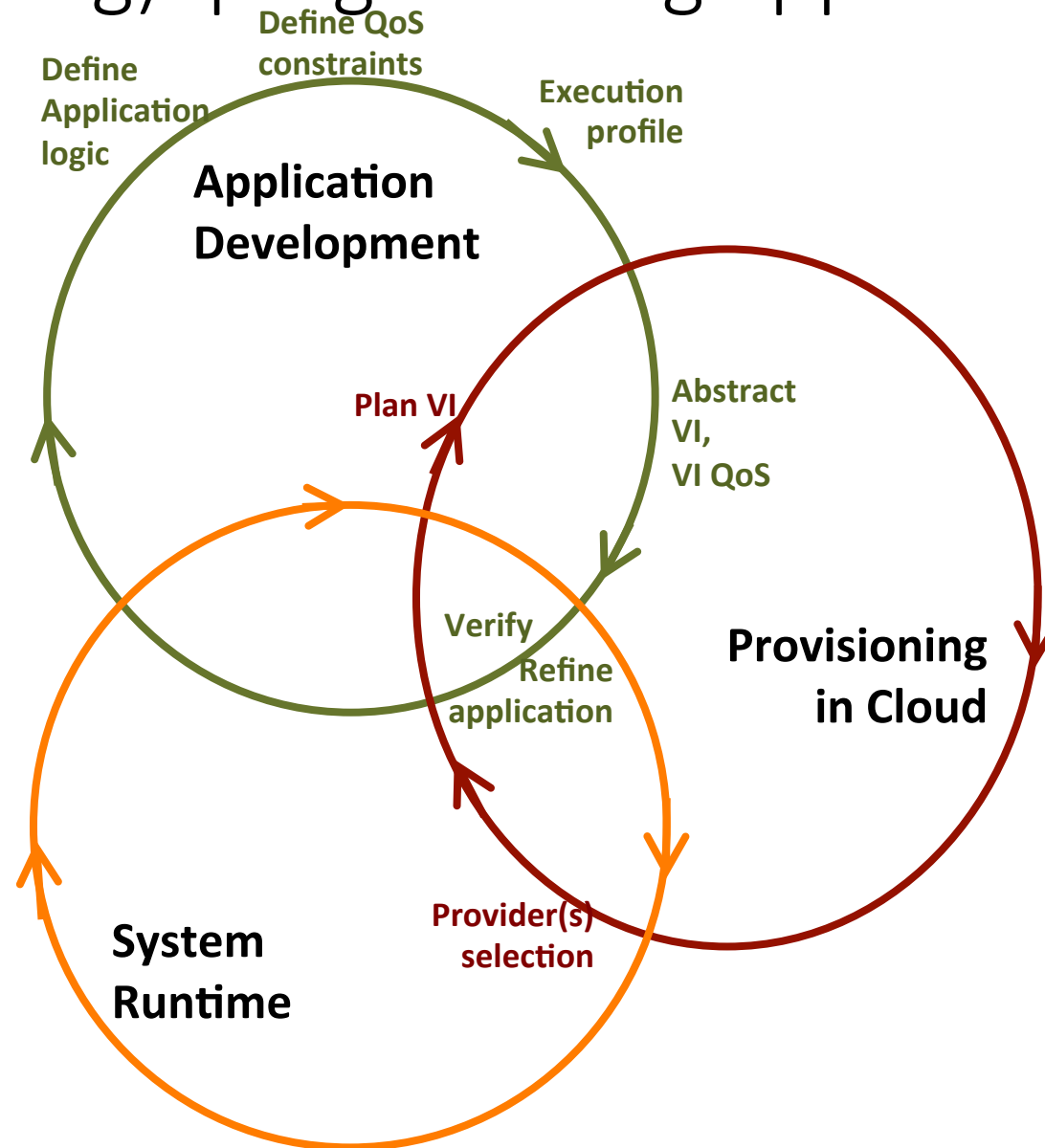


# SIDE(Co-programming): programming application-infrastructure

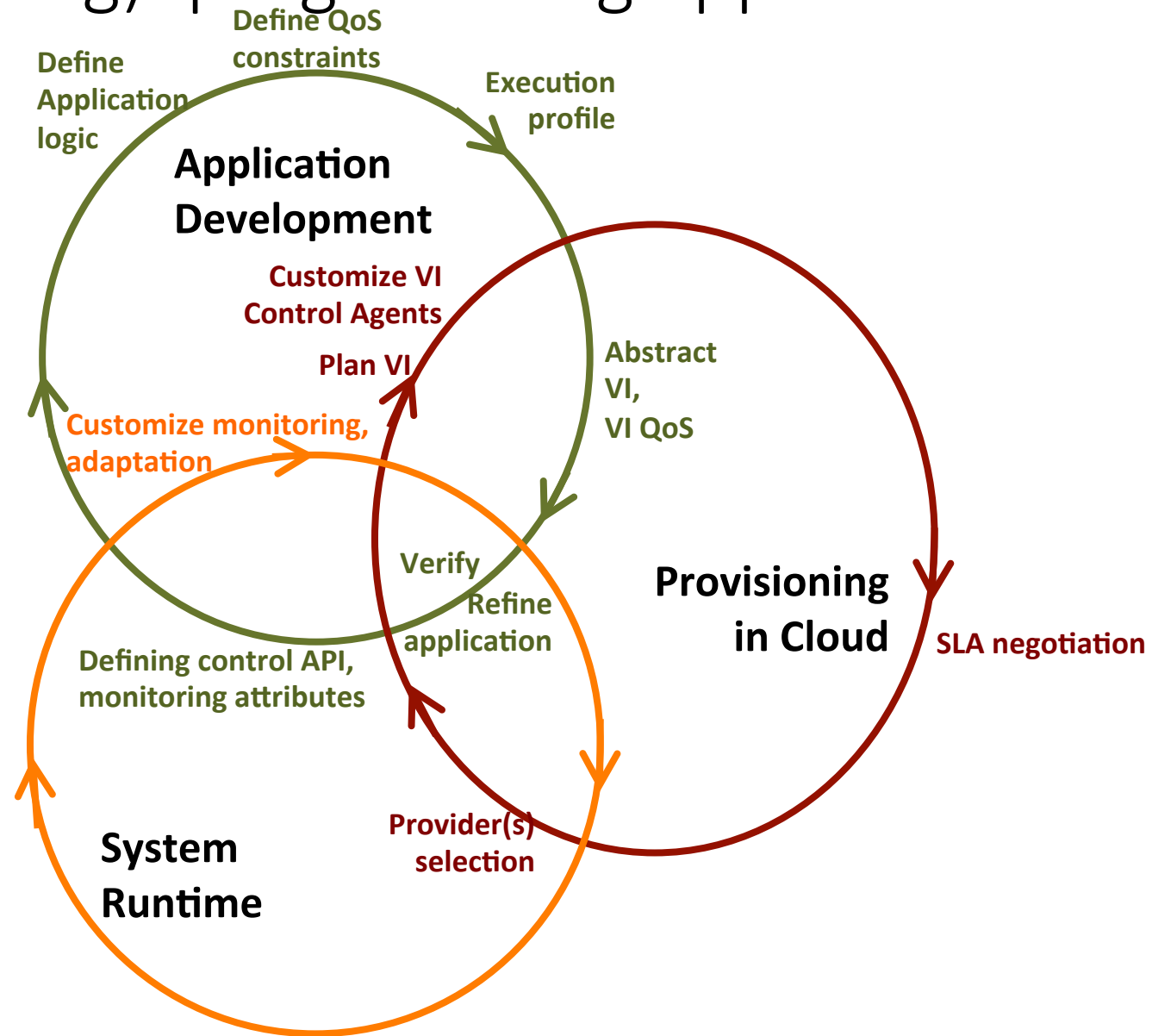


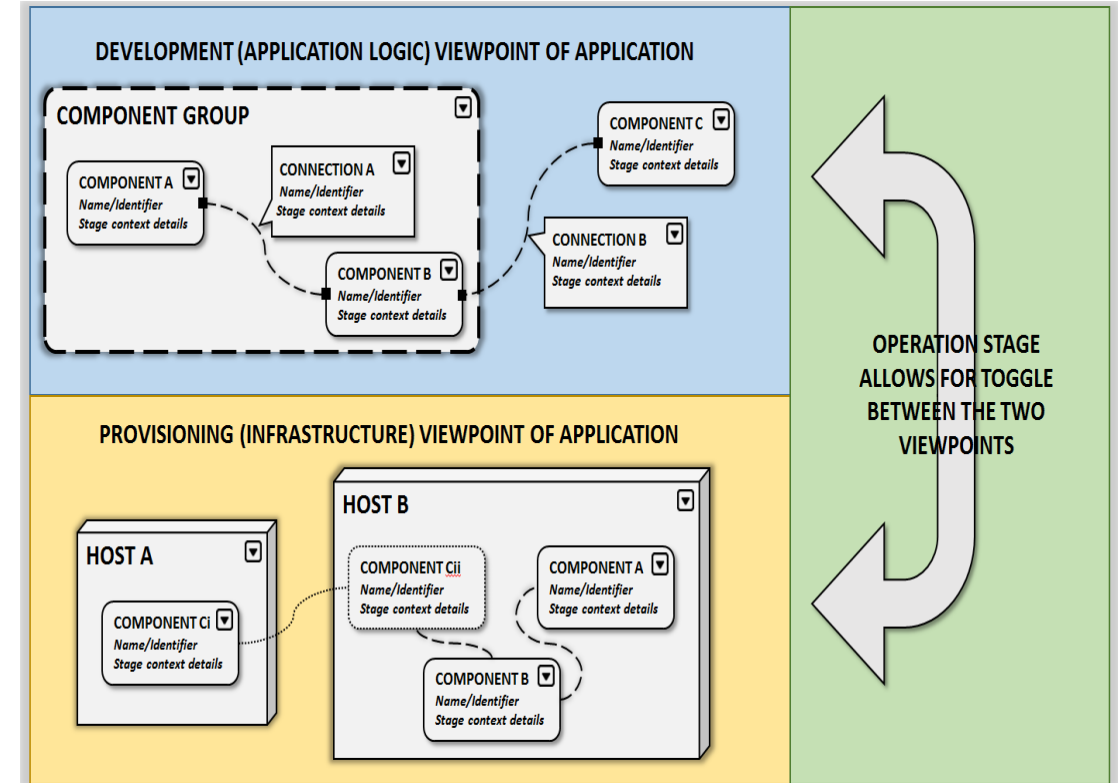
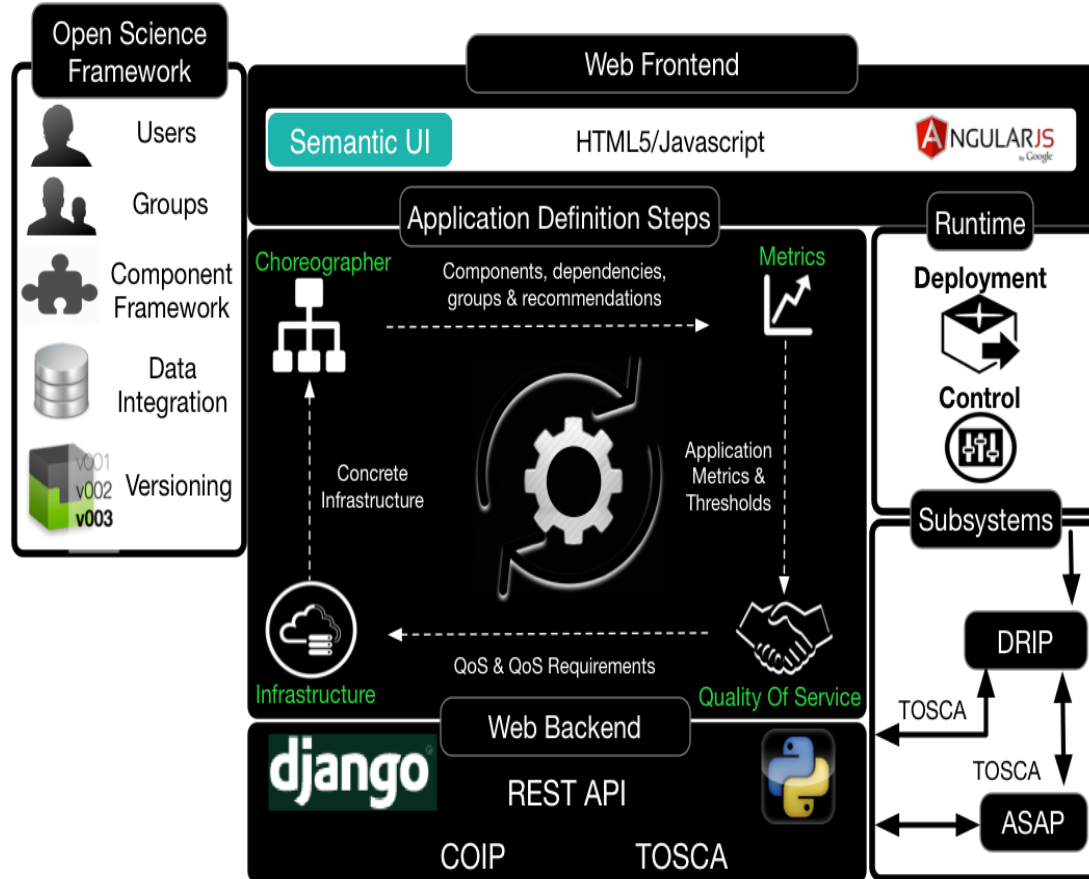


# SIDE(Co-programming): programming application-infrastructure



# SIDE(Co-programming): programming application-infrastructure



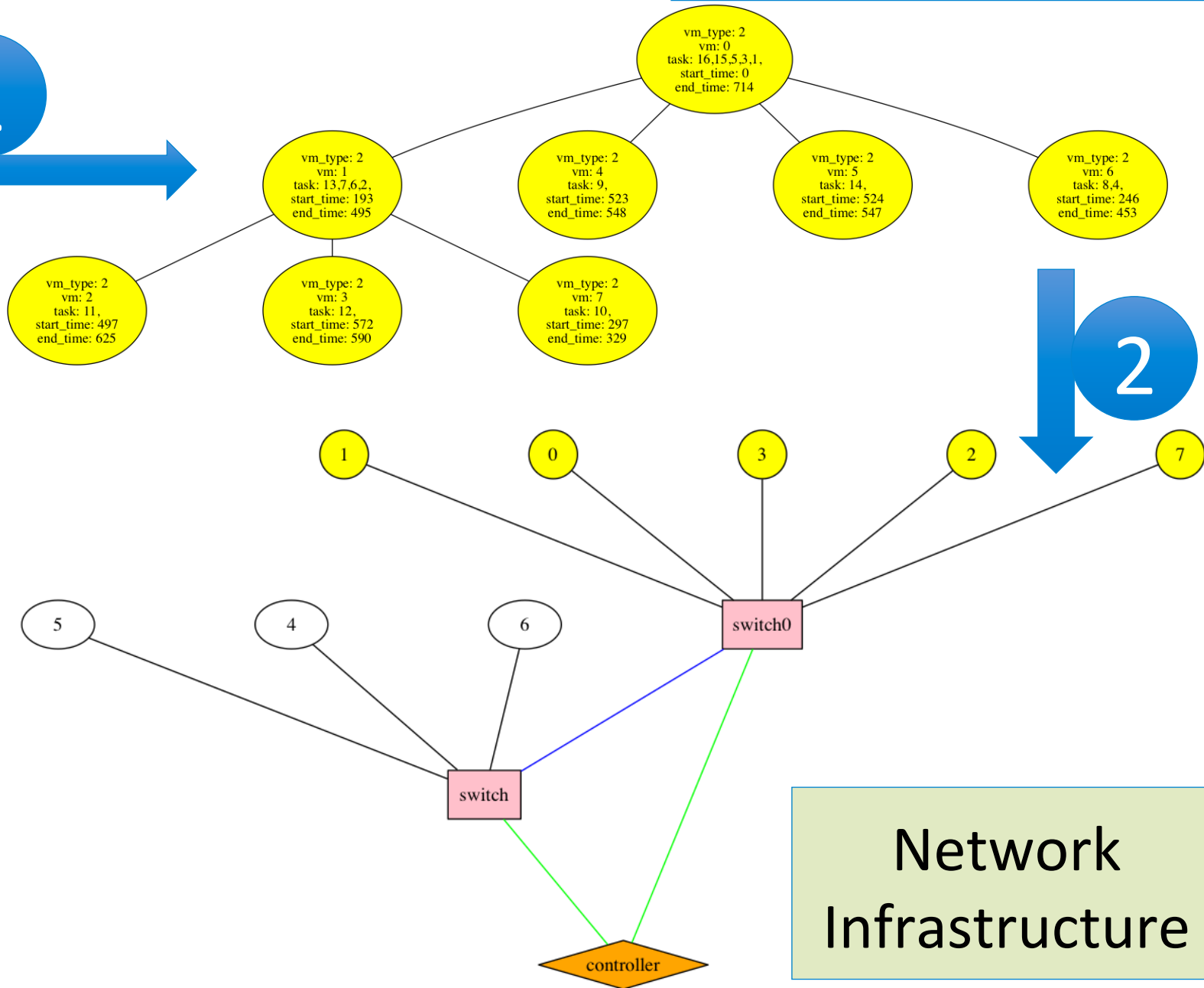
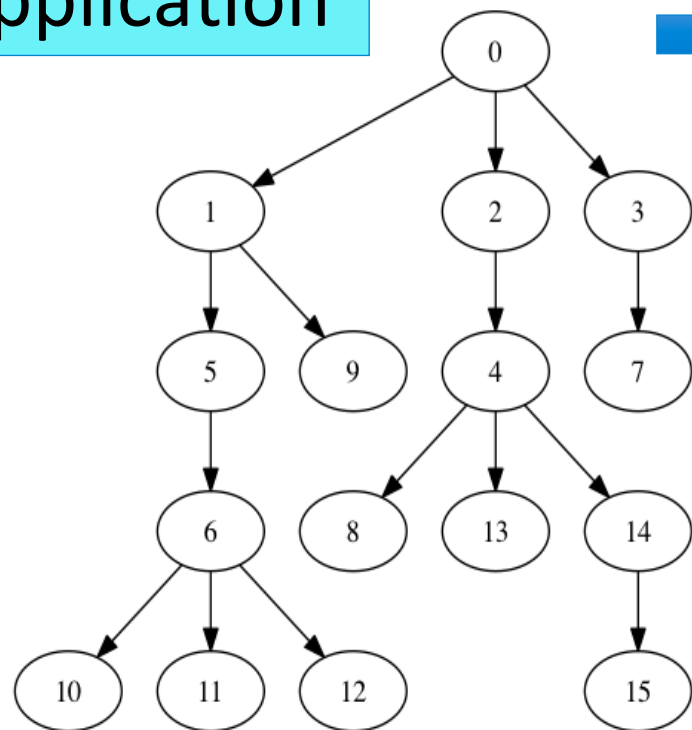


# Application $\leftrightarrow$ Infrastructure

VM Infrastructure

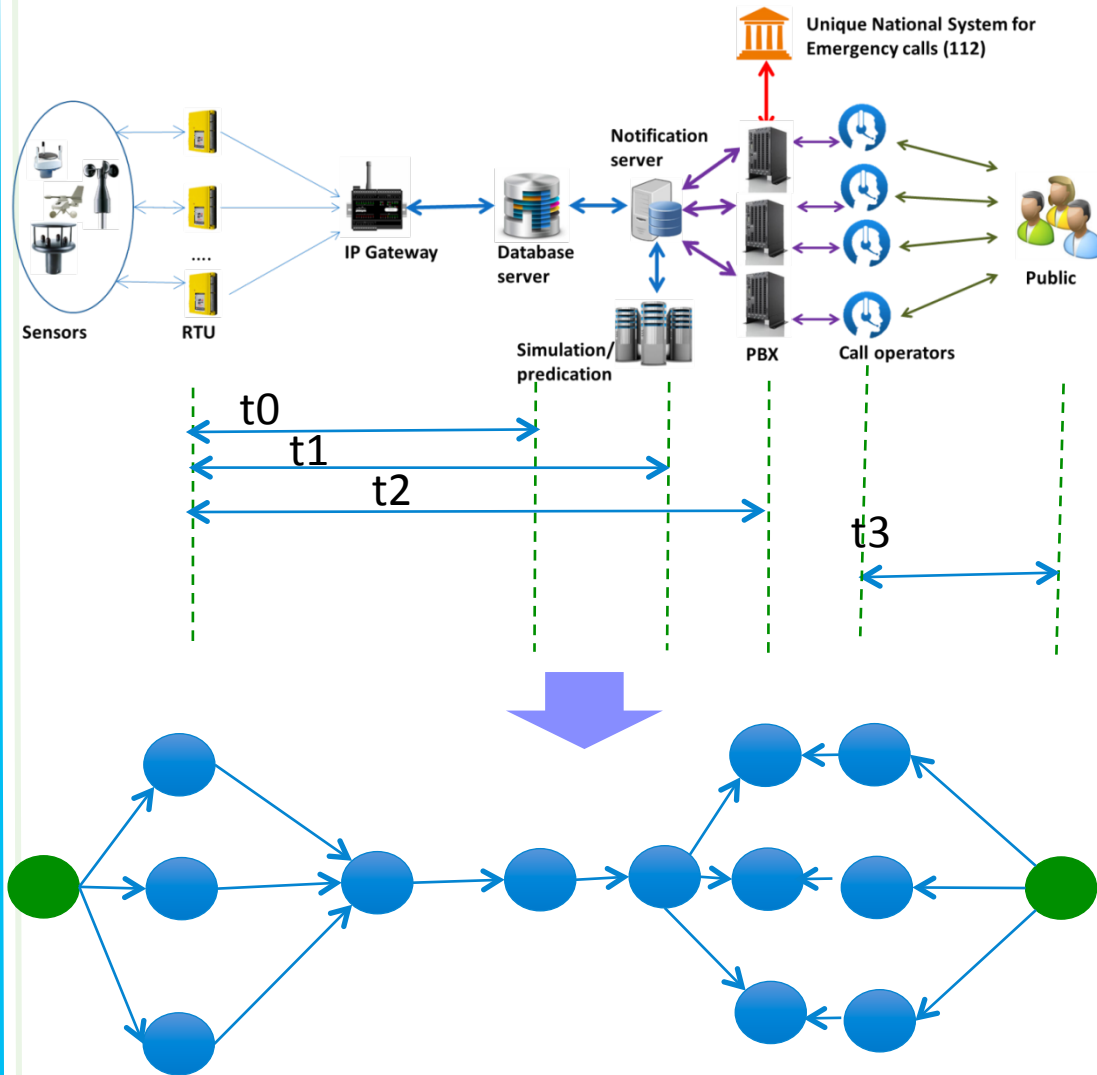
Application

1

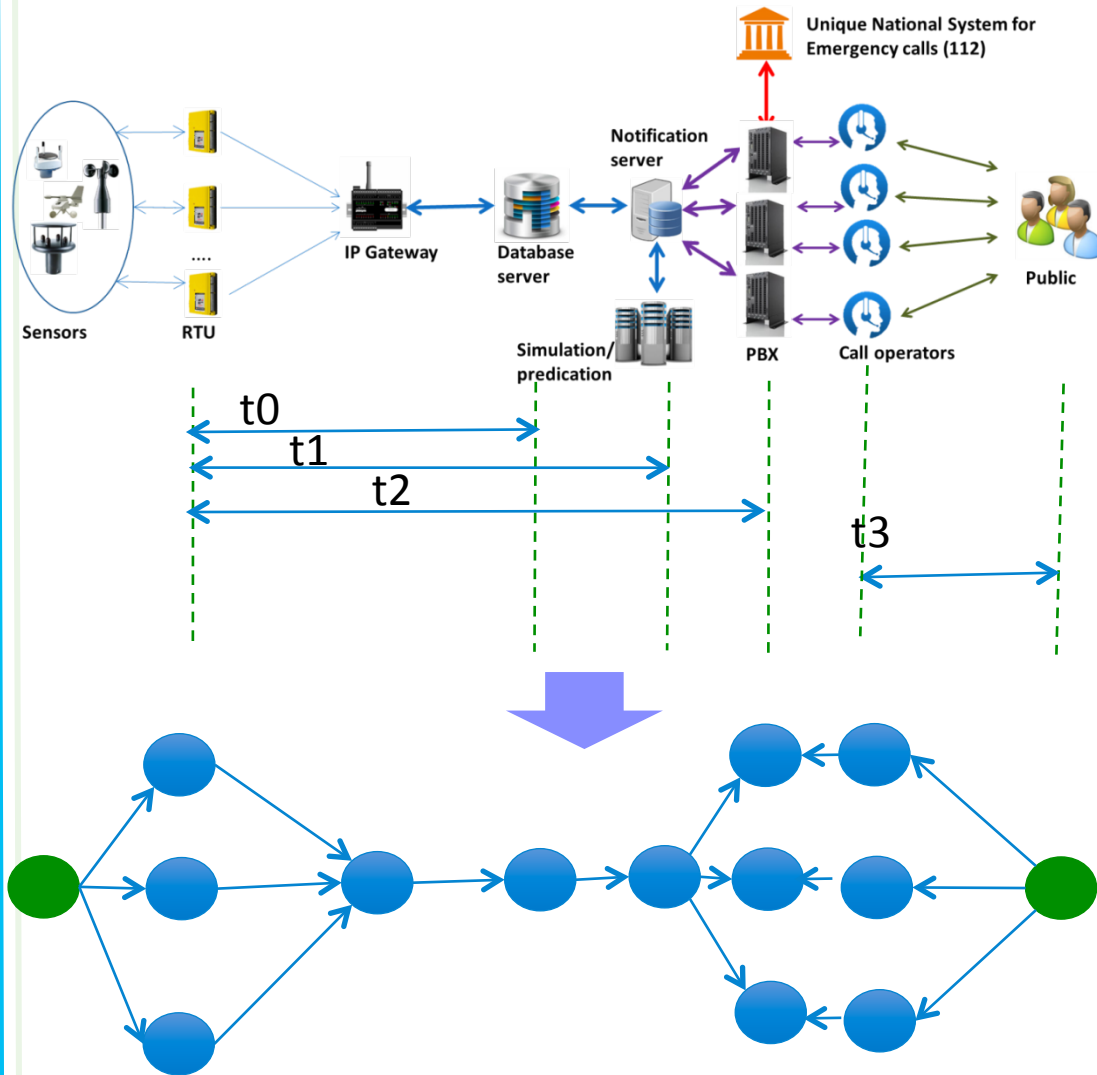


2

Network Infrastructure

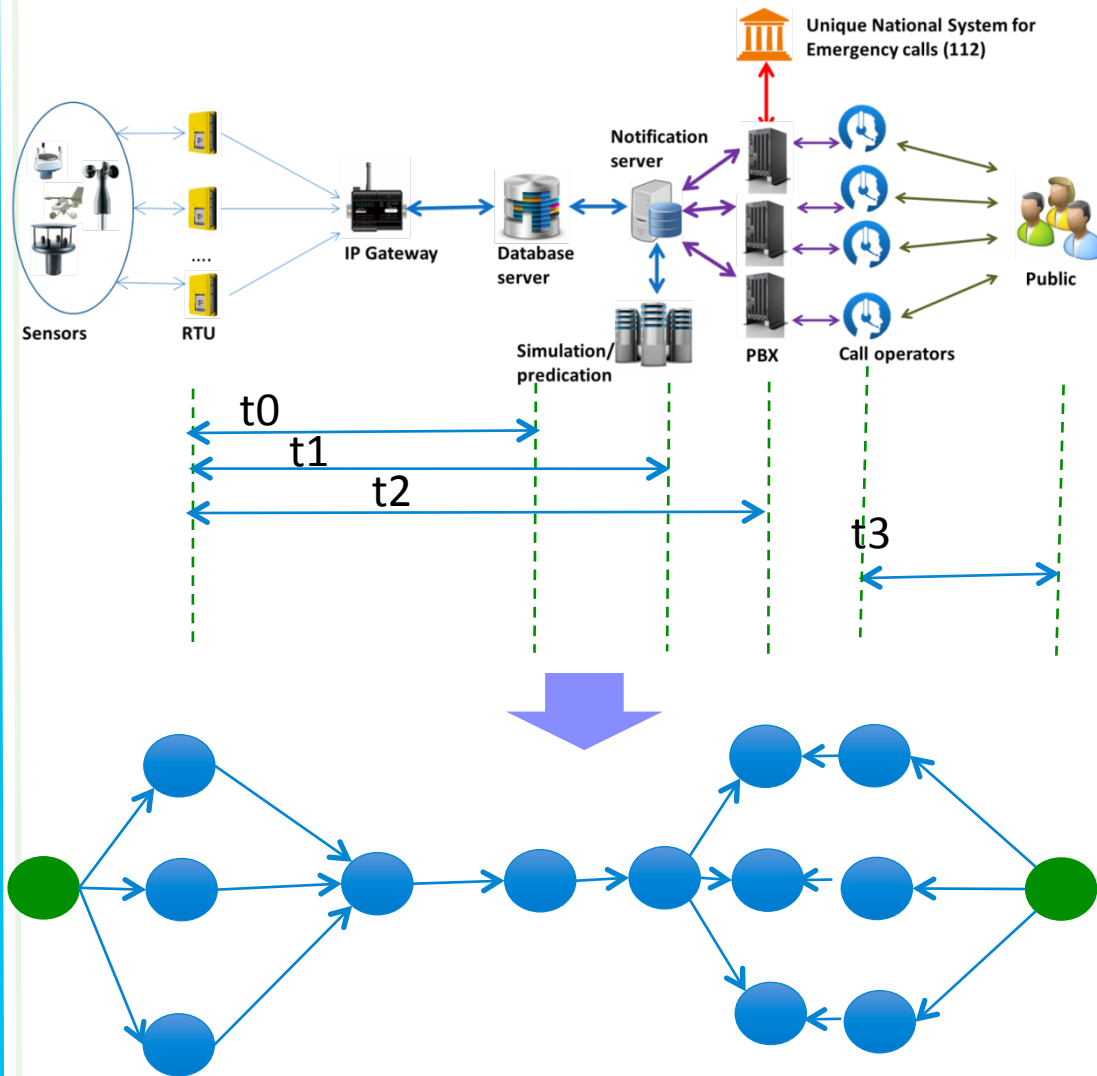


- Model the application as DAG.
  - Processes can be temporal with clear life time
  - Persistent services
- Deadlines
  - t0: the sensor data collection has to be finished
  - t1: the decision making procedure of the event has to be finished
  - t2: actions after the decision have to be finished
  - t3: the response time for the coming call

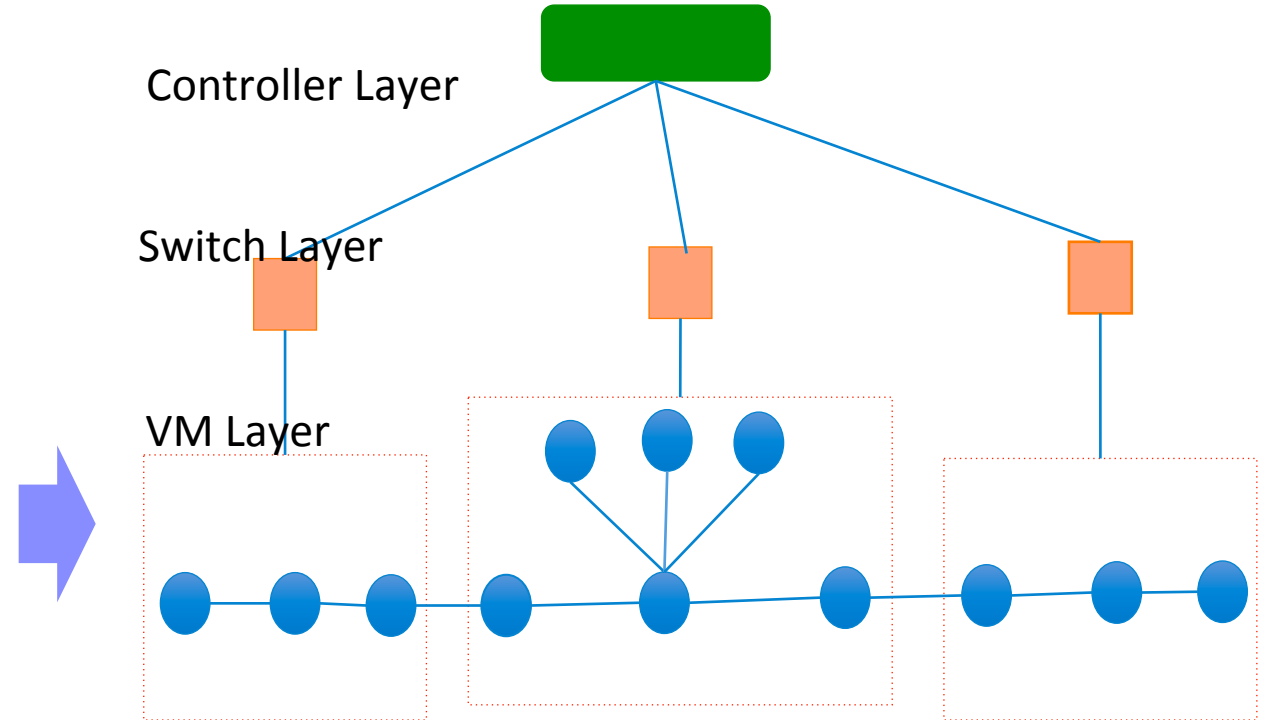


- Select suitable VM for each task (service), so that
  - The finishing time of each node is within the range of the deadline
  - Tasks are grouped to reduce communication cost
  - The total cost of the VMs are minimal
- Pre-condition:
  - The performance of a task on a specific type of VM is known
- IC-PCP algorithm based implementation
  - For multi deadlines

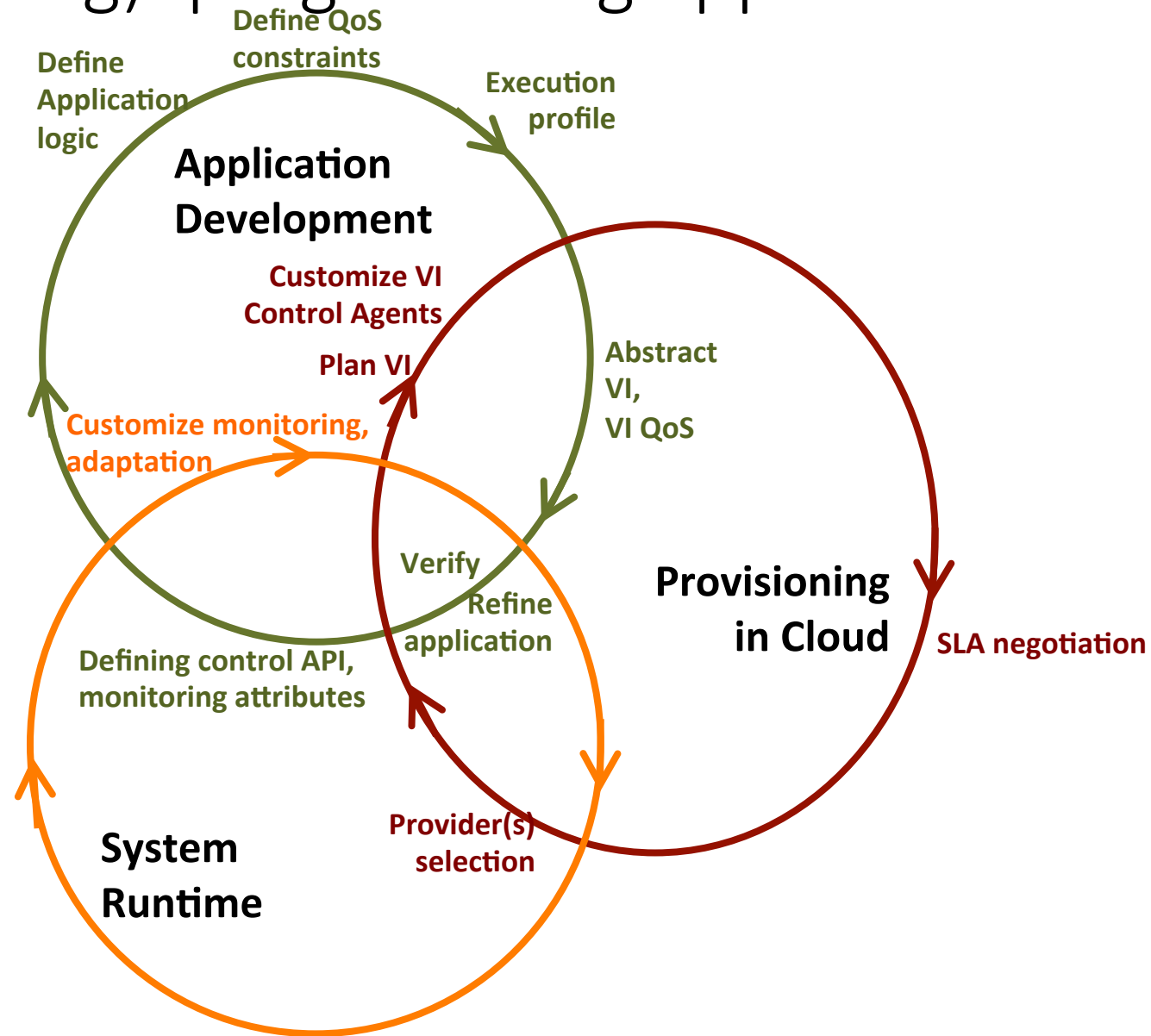




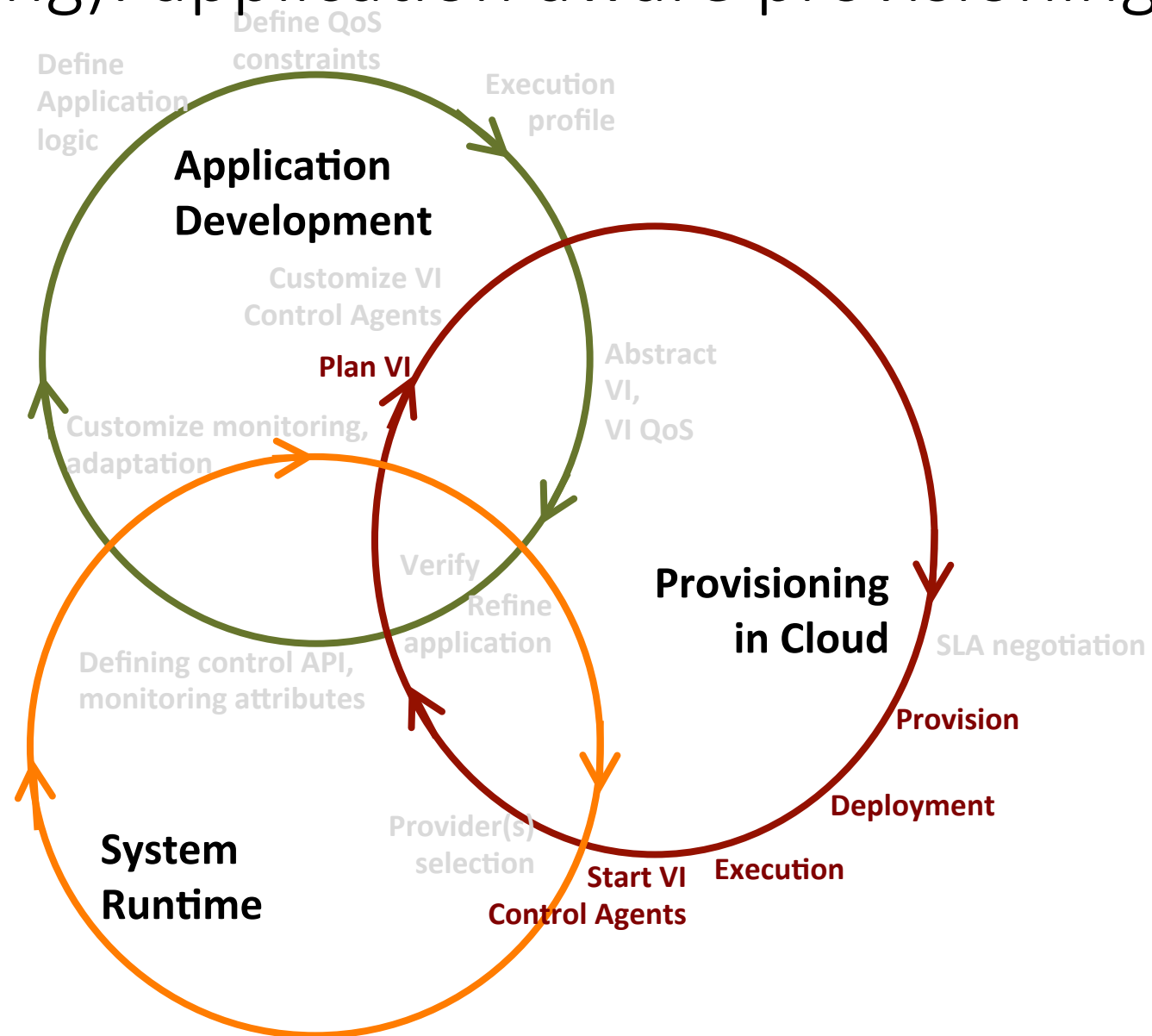
- Define the topology based on the VM based on the application logic and the critical paths
- Place virtual network switch and controller
- Related work



# SIDE(Co-programming): programming application-infrastructure



# DRIP(Co-provisioning): application aware provisioning





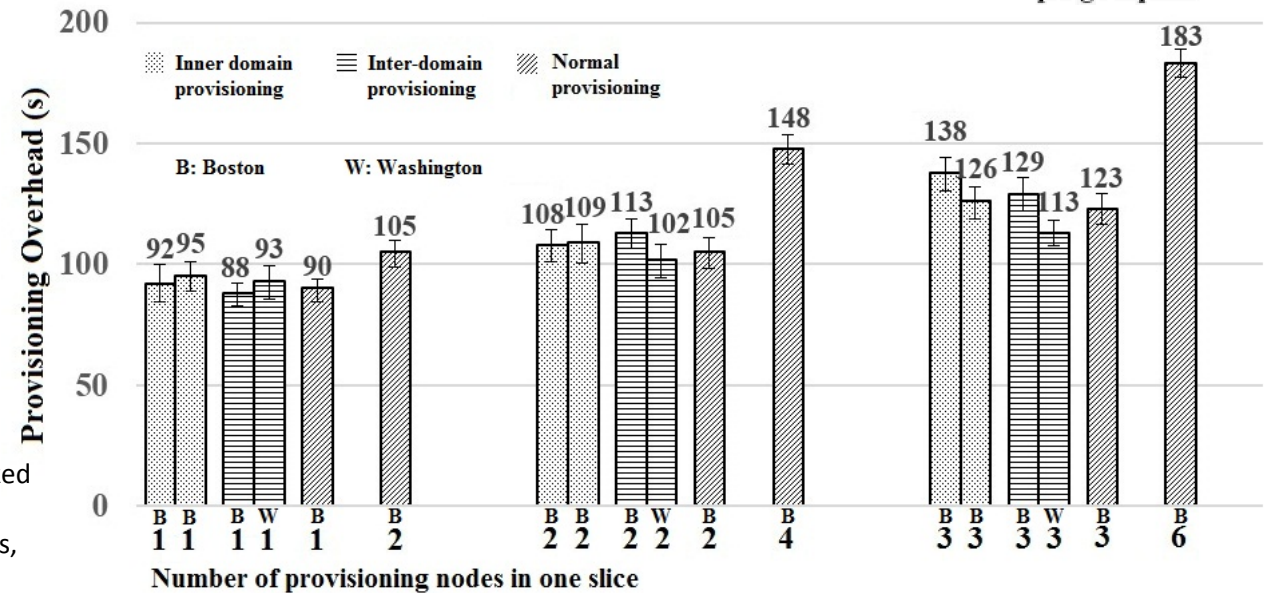
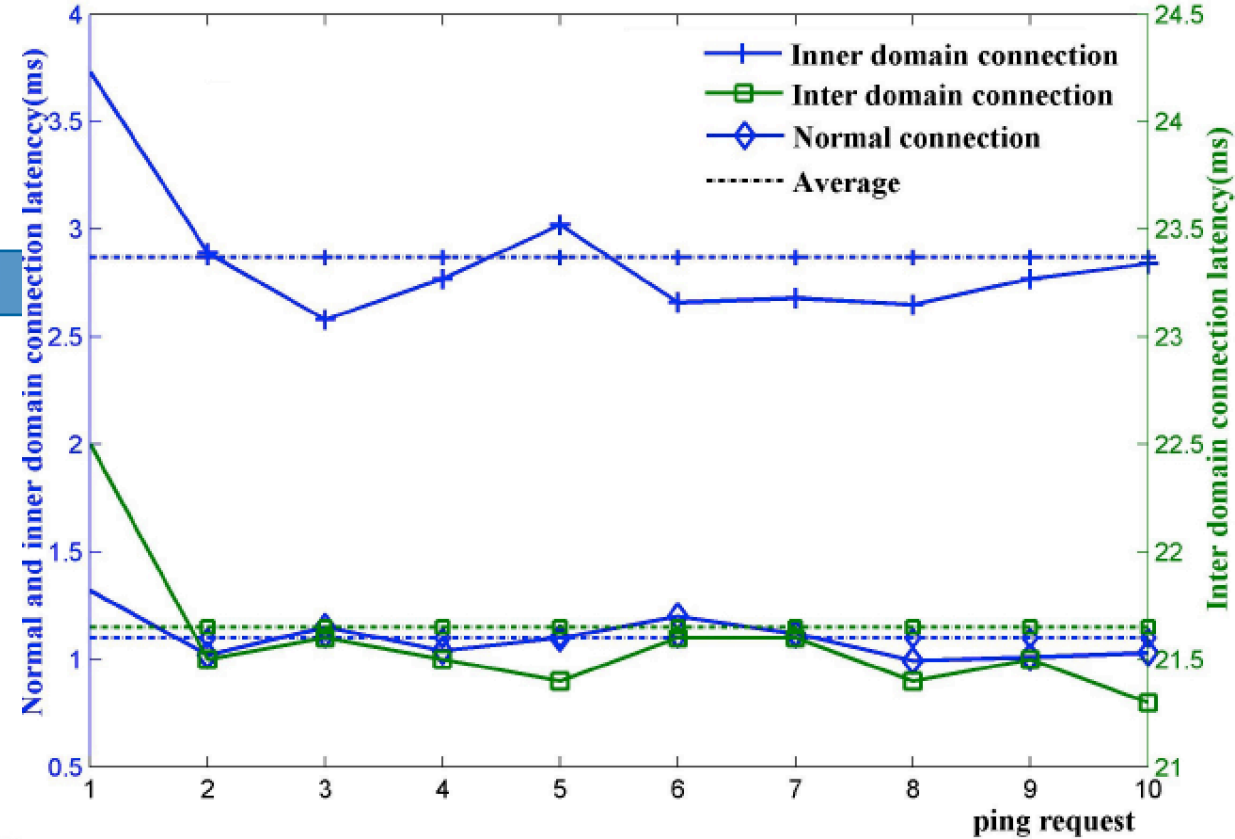
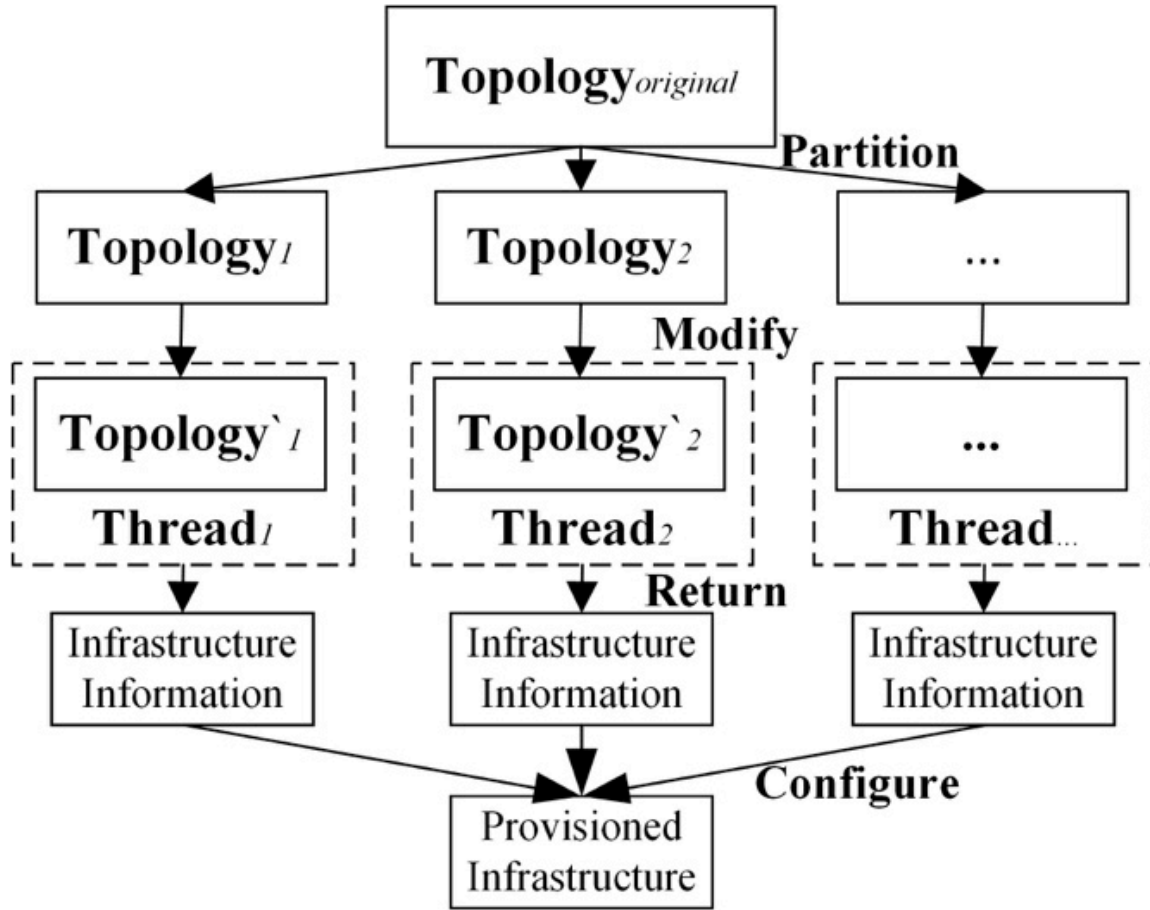
# DRIP (co-provisioning): provisioning based on application constraints



- **Provisioning agent:** efficiently provision complex virtual infrastructures on multiple data centres based on data flow, geo-location, and price etc.
- **Deployment agent:** efficiently schedule the software deployment based on application topology
  
- **Step 1:** decompose the planned virtual infrastructure based on selected data centres, locations of the sensors, or other constraints
- **Step 2:** provision in parallel the virtual infrastructure on different data centres (providers)
- **Step 3:** set up the network connectivity and keep it transparent
- **Step 4:** schedule the deployment based on execution sequence and data dependencies



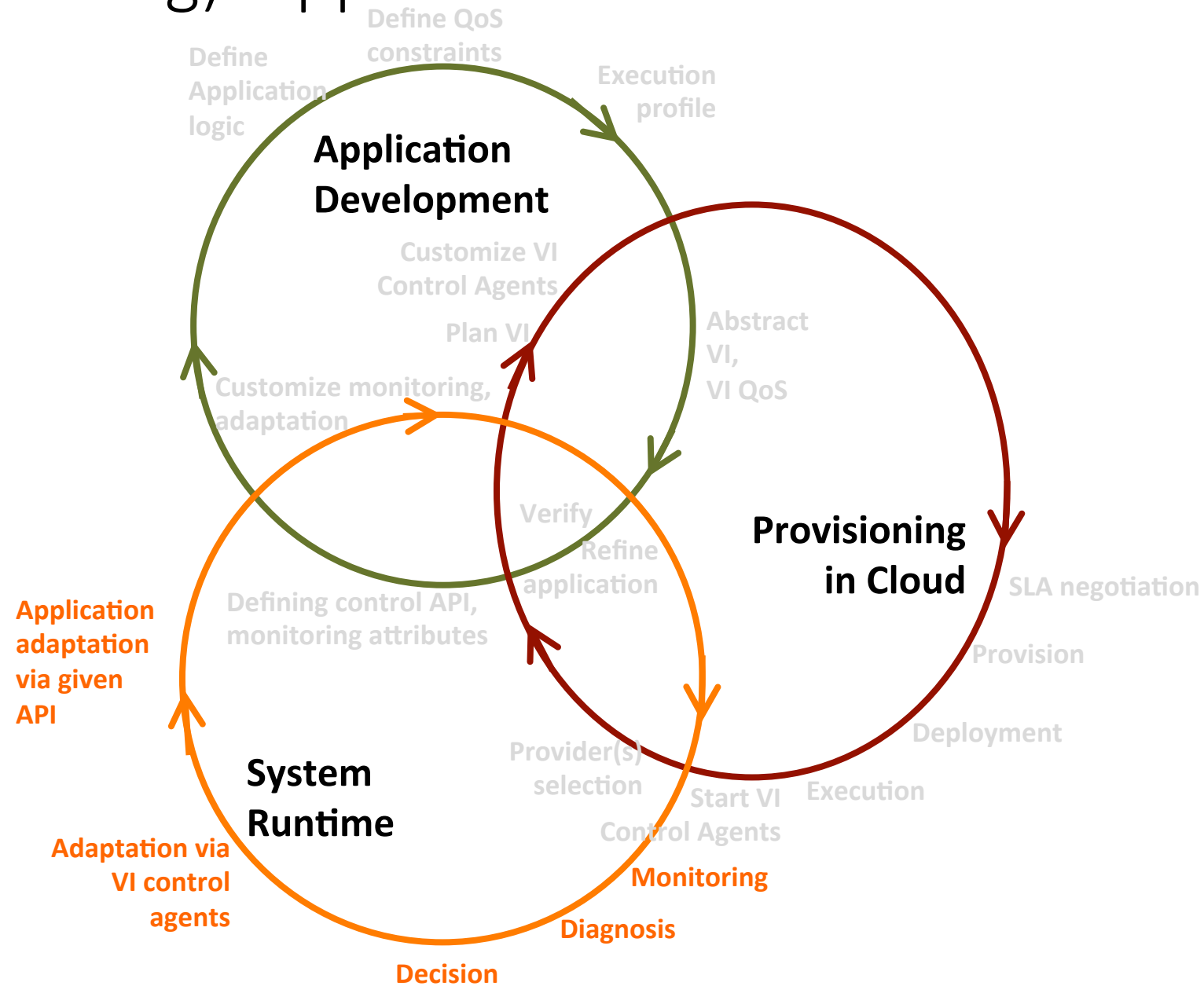
# Basic approach



Zhou, H., et al., (2016) Fast Resource Co-provisioning for Time Critical Application Based on Networked Infrastructure, IEEE International Conference on CLOUD (CLOUD) 2016, San Francisco US.

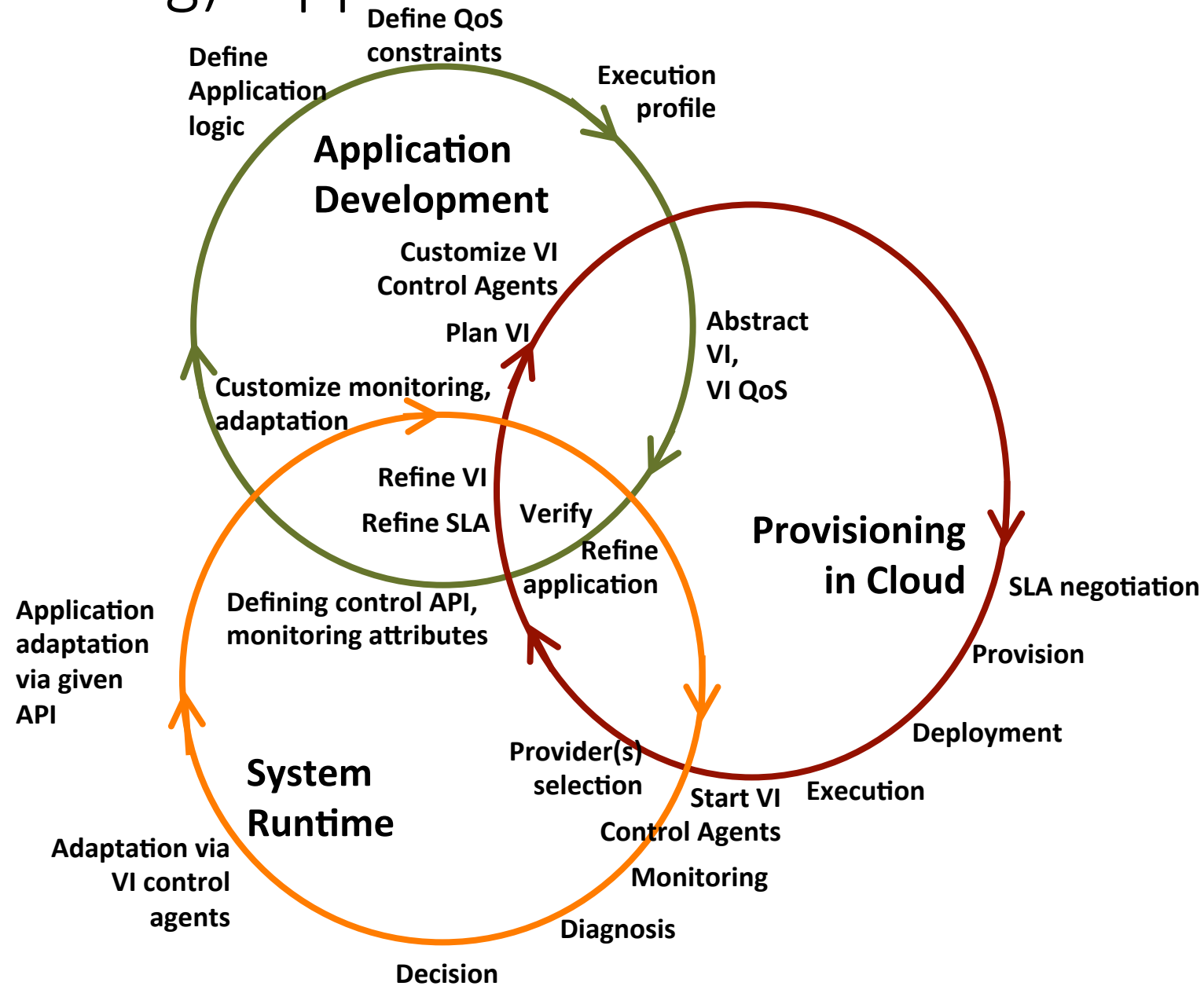
Zhou, H., et al., (2016) Fast and Dynamic Resource Provisioning for Quality Critical Cloud Applications, IEEE International Symposium On Real-Time Computing (ISORC) 2016, York UK

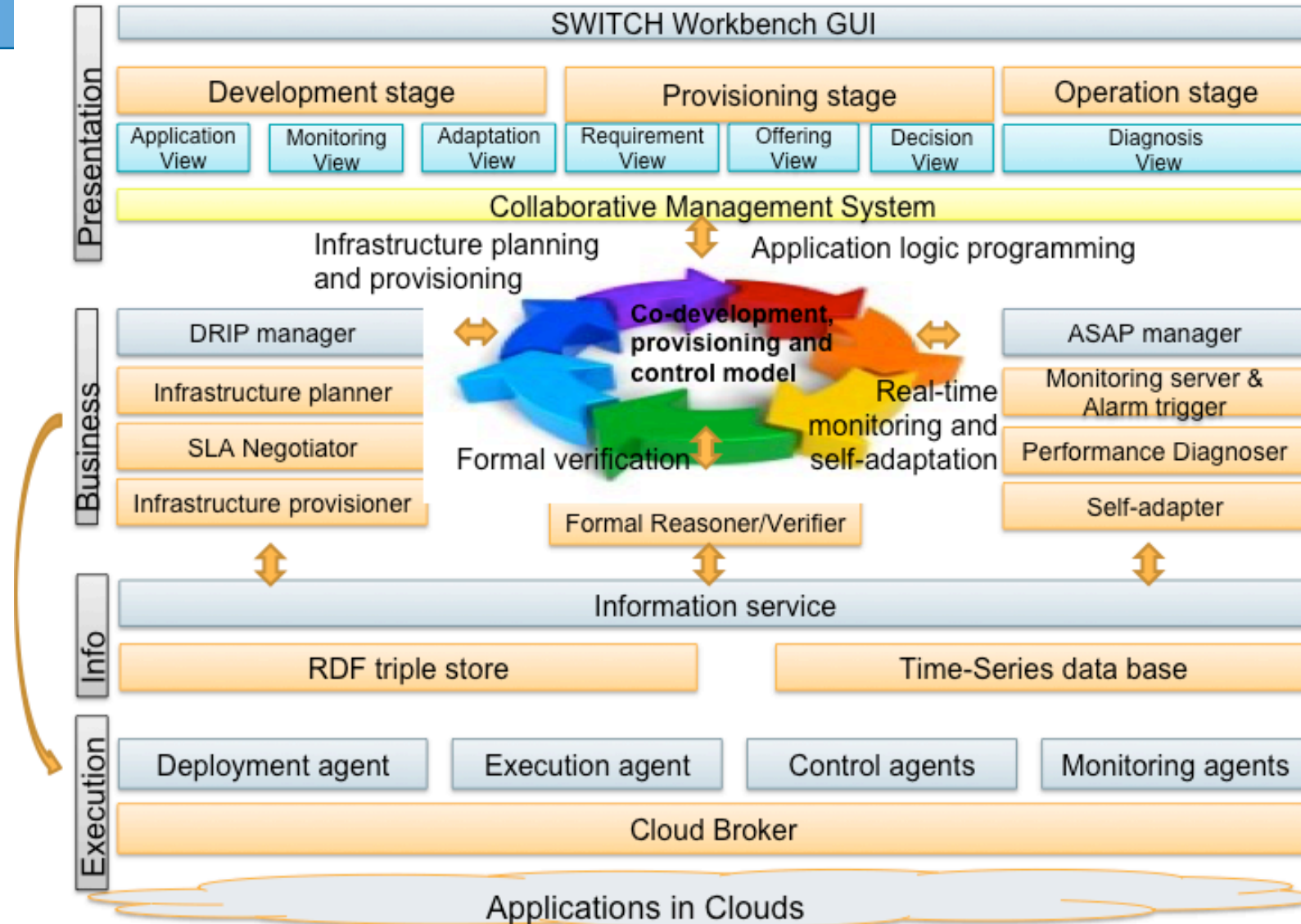
# ASAP(Co-controlling): application infrastructure co-controlling





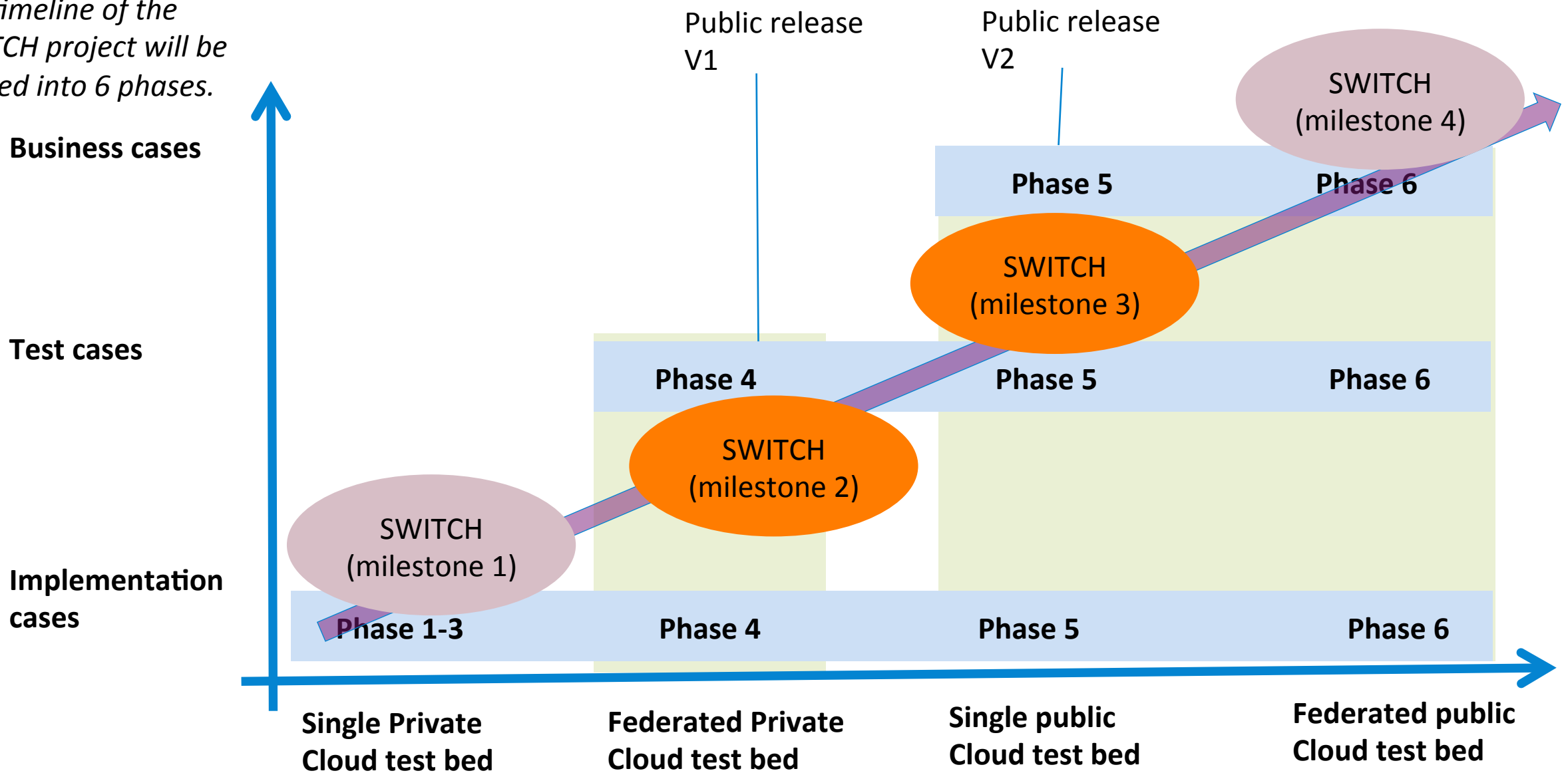
# ASAP(Co-controlling): application infrastructure co-controlling



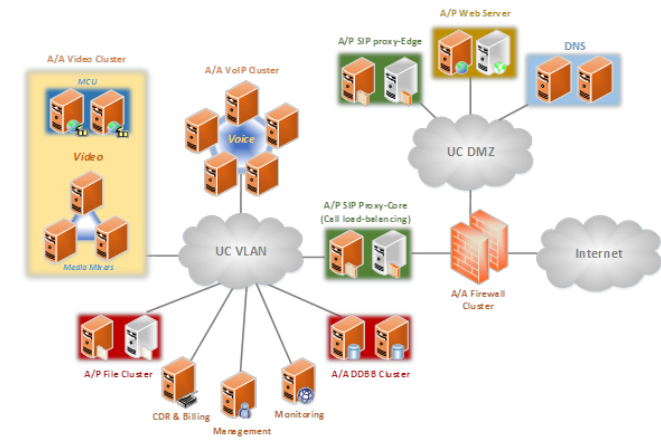
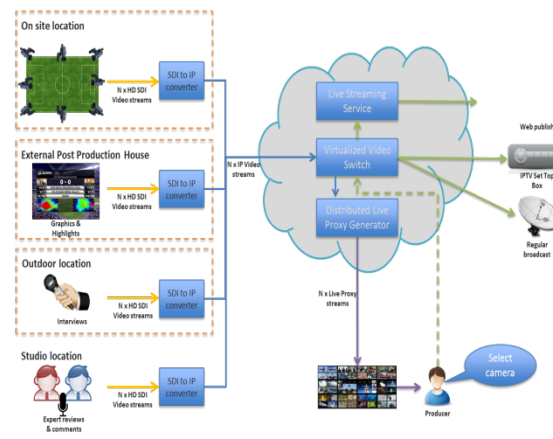
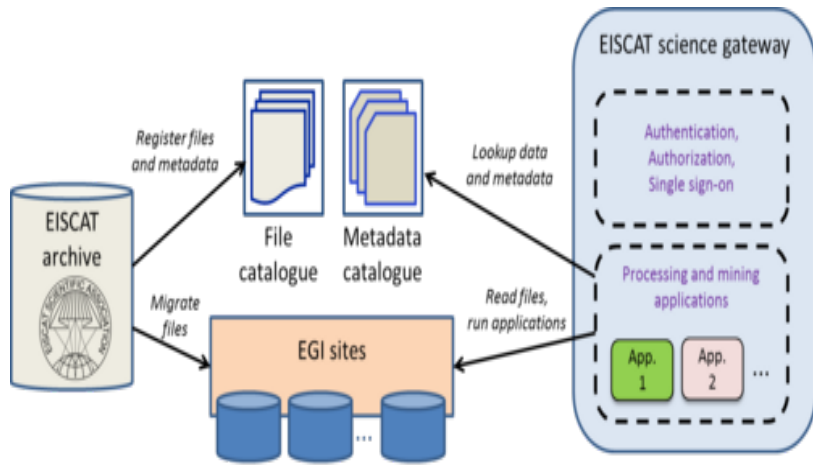
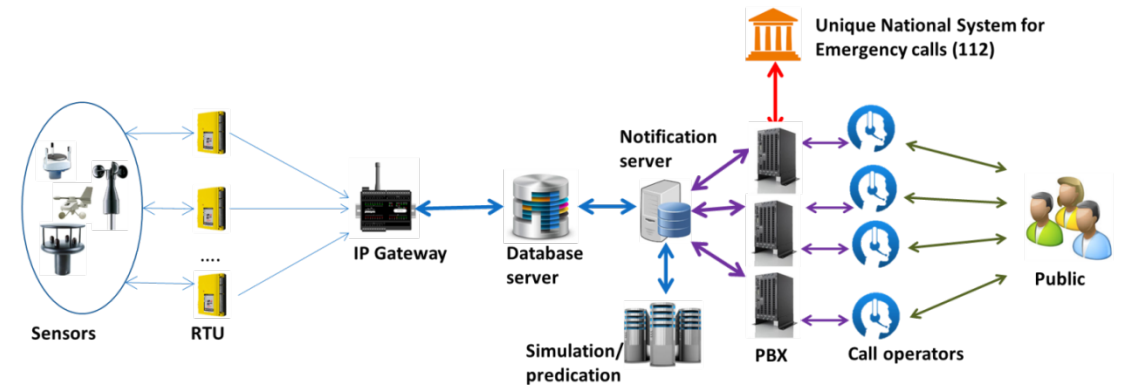


# Development plan

*The timeline of the SWITCH project will be divided into 6 phases.*



- Time critical Cloud applications
  - Business collaboration platforms,
  - Disaster early warning
  - Live event broadcasting
- Research infrastructures: ENVRIPlus, VRE4EIC
- Decision making, IoT, SmartCity etc.





# Summary



- Software engineering approaches and tools are urgently needed for time critical cloud applications. A workbench solution provides support at different phase of the application lifecycle
- Cloud environments provide elastic and programmable resources for time critical applications; however, effective solutions are needed for integrating these features in the programming and control model of the applications

- 2<sup>nd</sup> IT4RIs workshop: nearly real time data processing and time critical cloud applications (in the context of RTSS):
  - Time: **29/November**
  - Location: **Porto, Portage**
  - Web: <https://staff.fnwi.uva.nl/z.zhao/workshop/it4ris/>
- EU H2020, SWITCH: [www.switchproject.eu](http://www.switchproject.eu)



Zhiming Zhao



Carlos Rodrigo Rubia  
Marcos



Andrew Jones



Vlado Stankovski



George Surciu



Pedro Ferreira



Cees de Laat



Ian Taylor



Alexandre viïsses