A Systems of Systems

perspective on

The Internet of Things

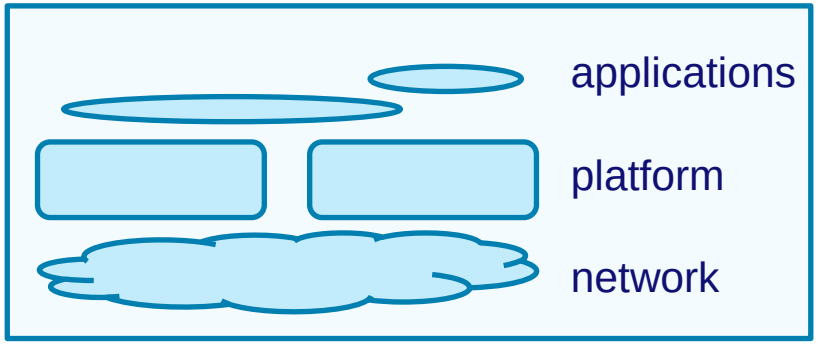
Johan Lukkien
Eindhoven University

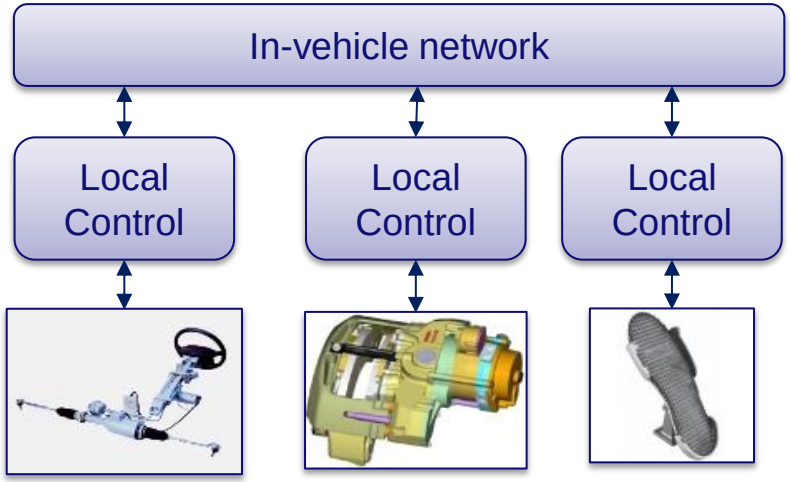
SAN

TU/e Technische Universiteit Eindhoven University of Technology

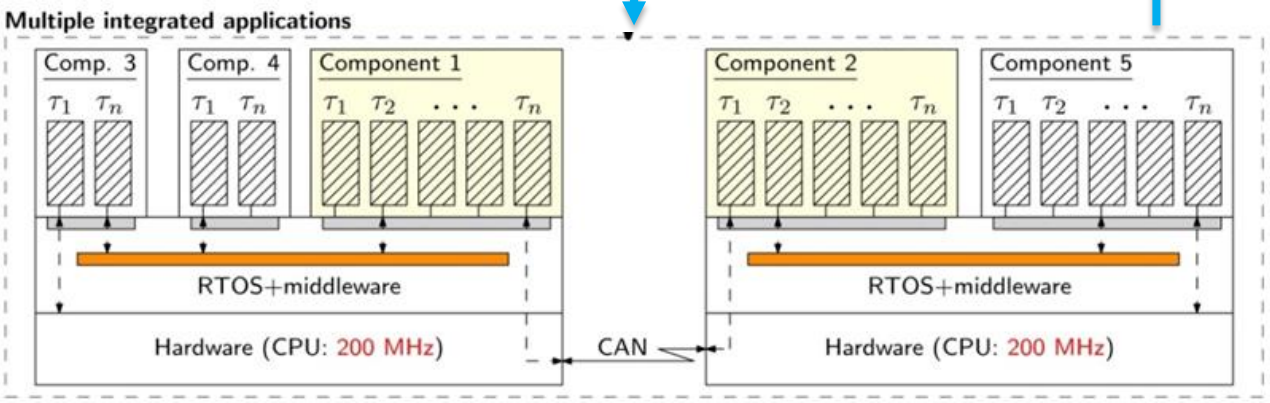
Where innovation starts

System

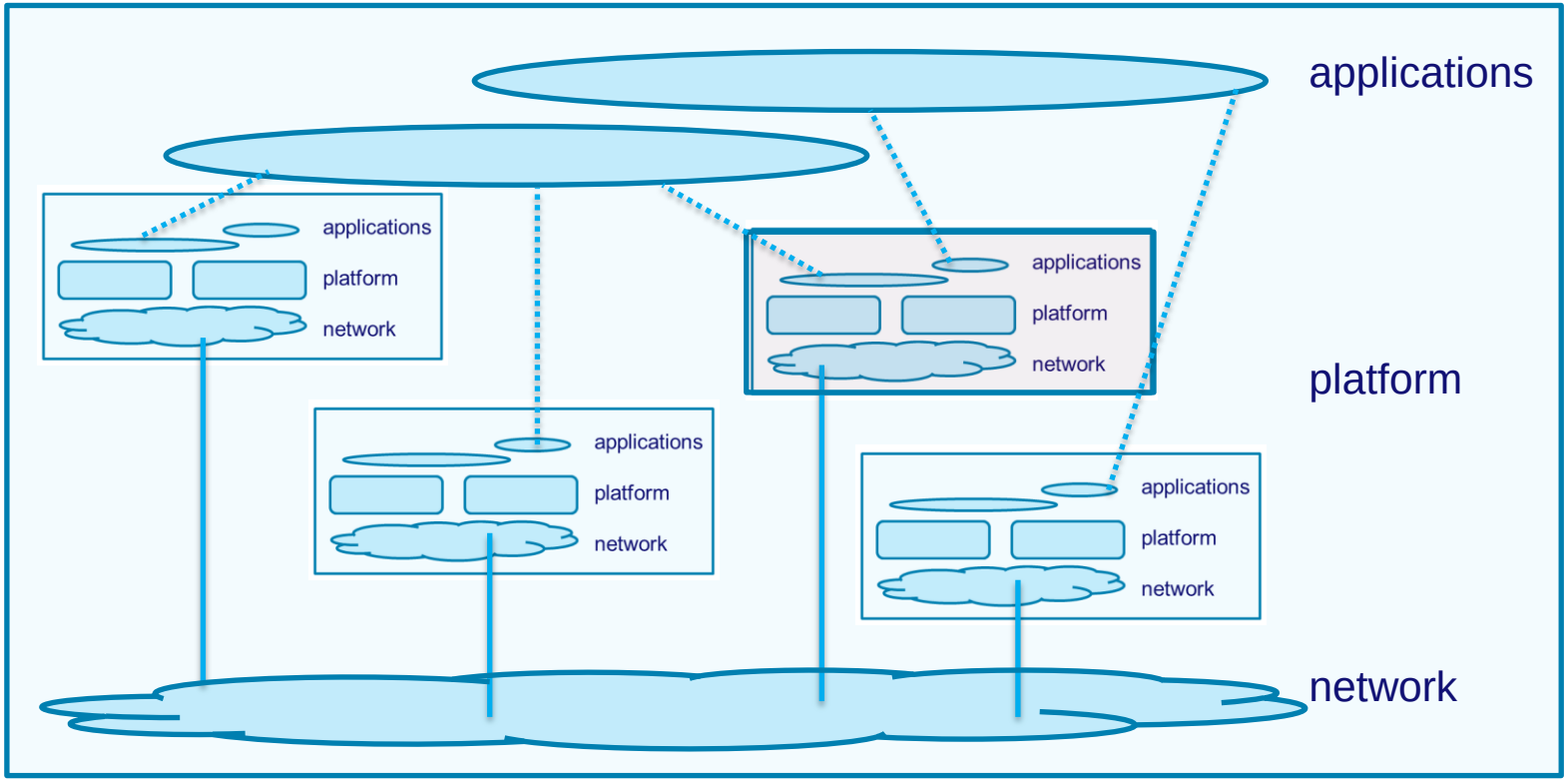


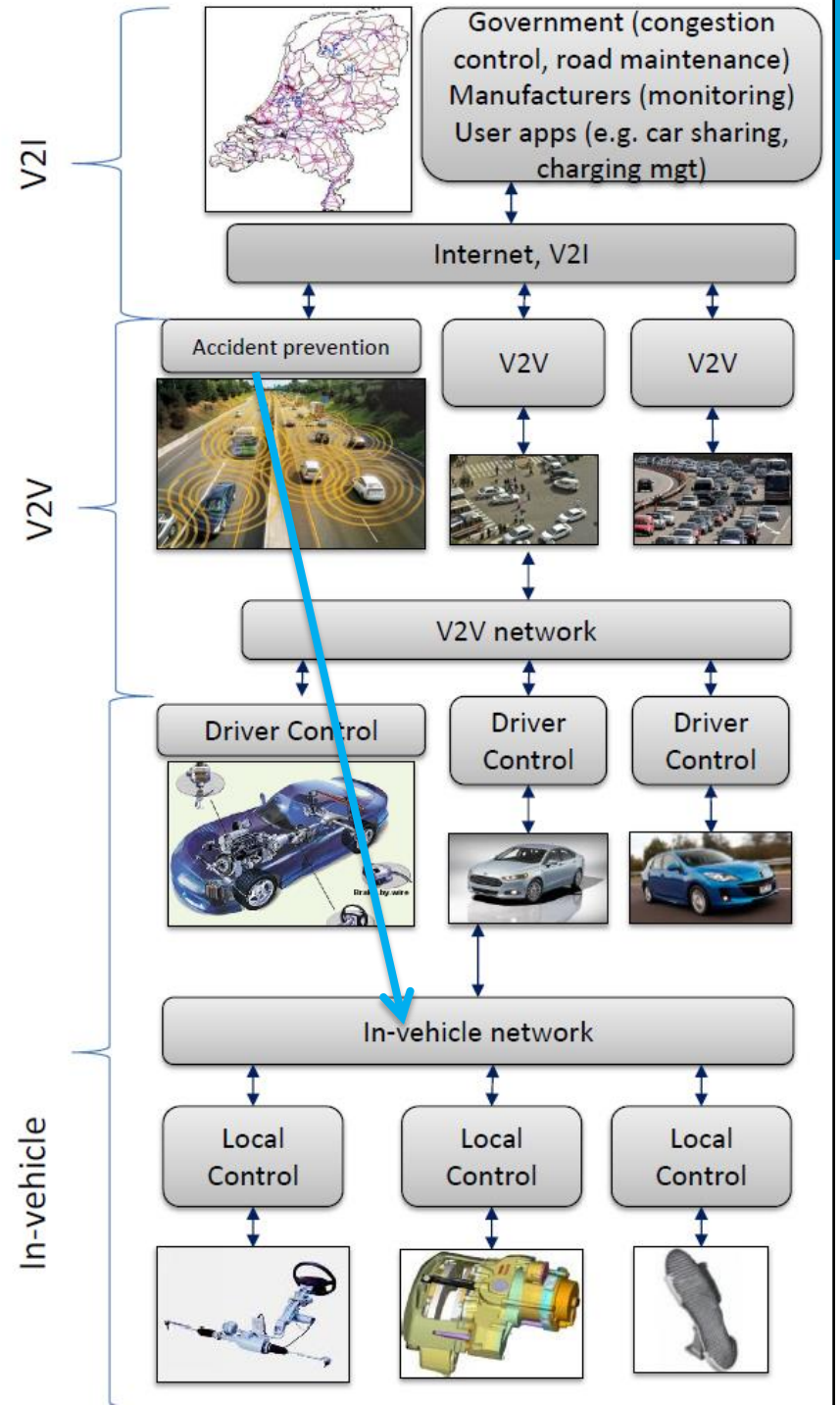


Reservations, Budgets
Admission test
(temporal) Isolation
Real-time interfaces



System of systems



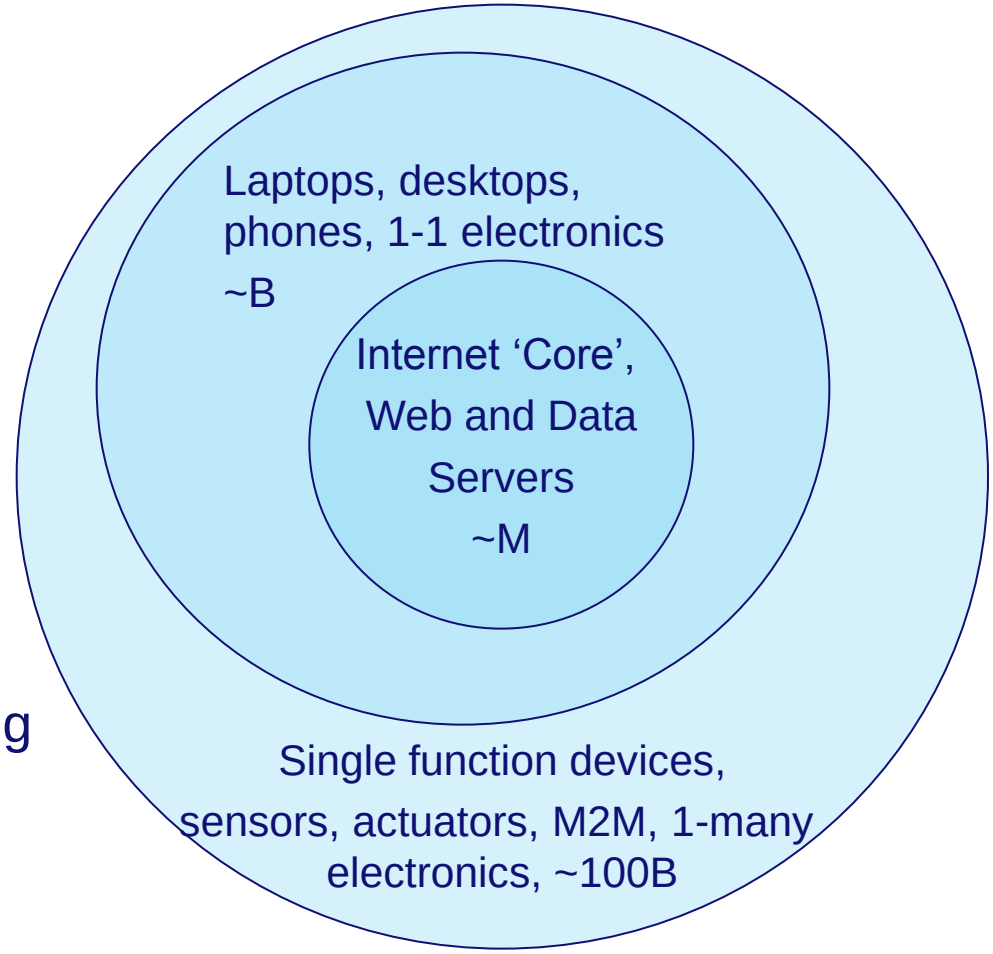


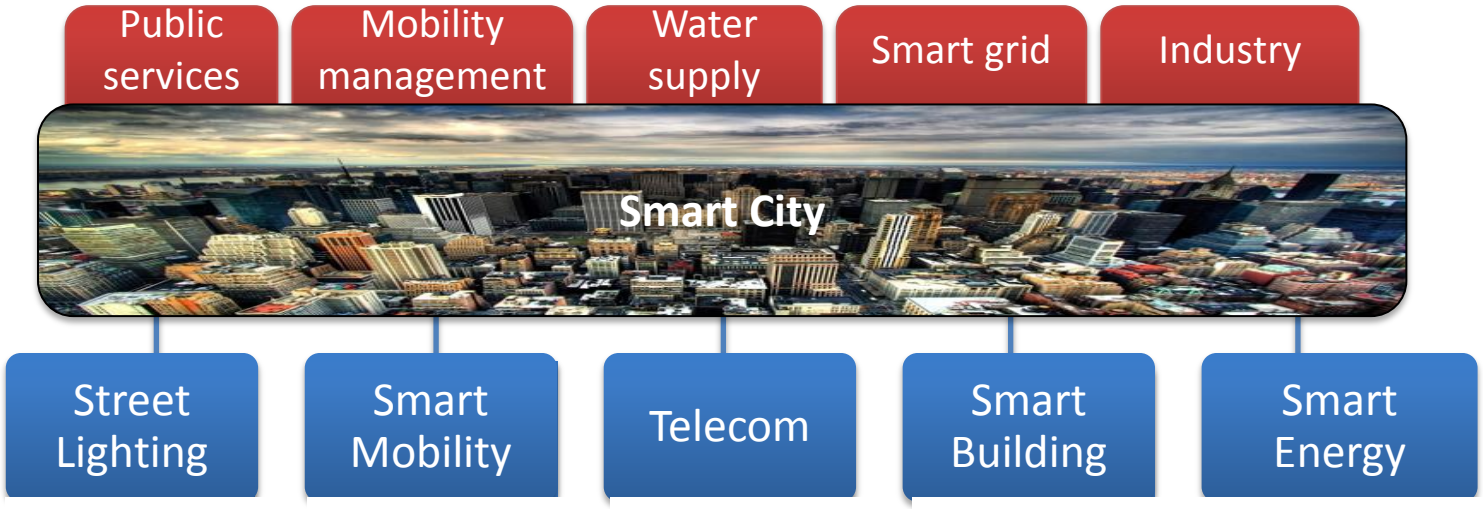
Intelligent Transportation Systems

- Vehicle: integrated system components
  - systems by themselves, but not independent
- V2V: system of systems (vehicles) with V2V applications
  - e.g. accident prevention, parking spot finding, collaborative driving
- V2I: 'slower' (higher latency) applications, global applications
- ITS is a SoS but also a typical IoT system (or Cyber Physical system)

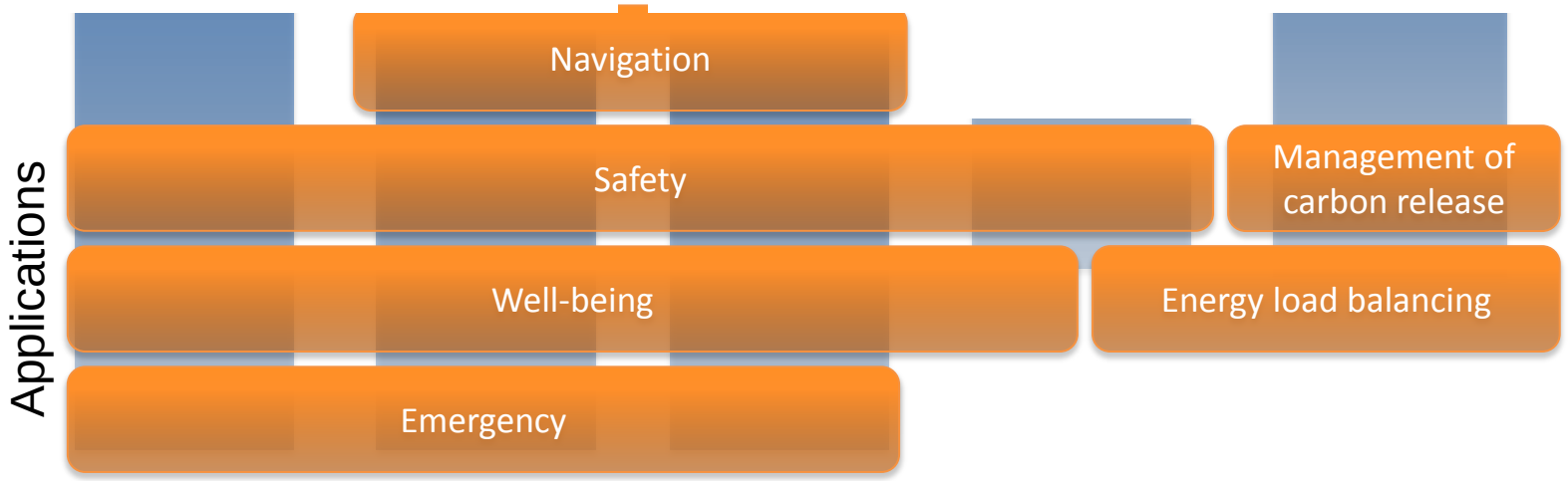
Internet of Things

- A unified protocol and naming scheme between every pair of devices
- Pervasive, extending network communication to billions of endpoints
- *Reaching into the physical world*, *reaching deep into (production) systems,* gathering large amounts of detailed information about states and events
- Moving into safety criticality (CPS)





Another typical SoS and IoT system(s)

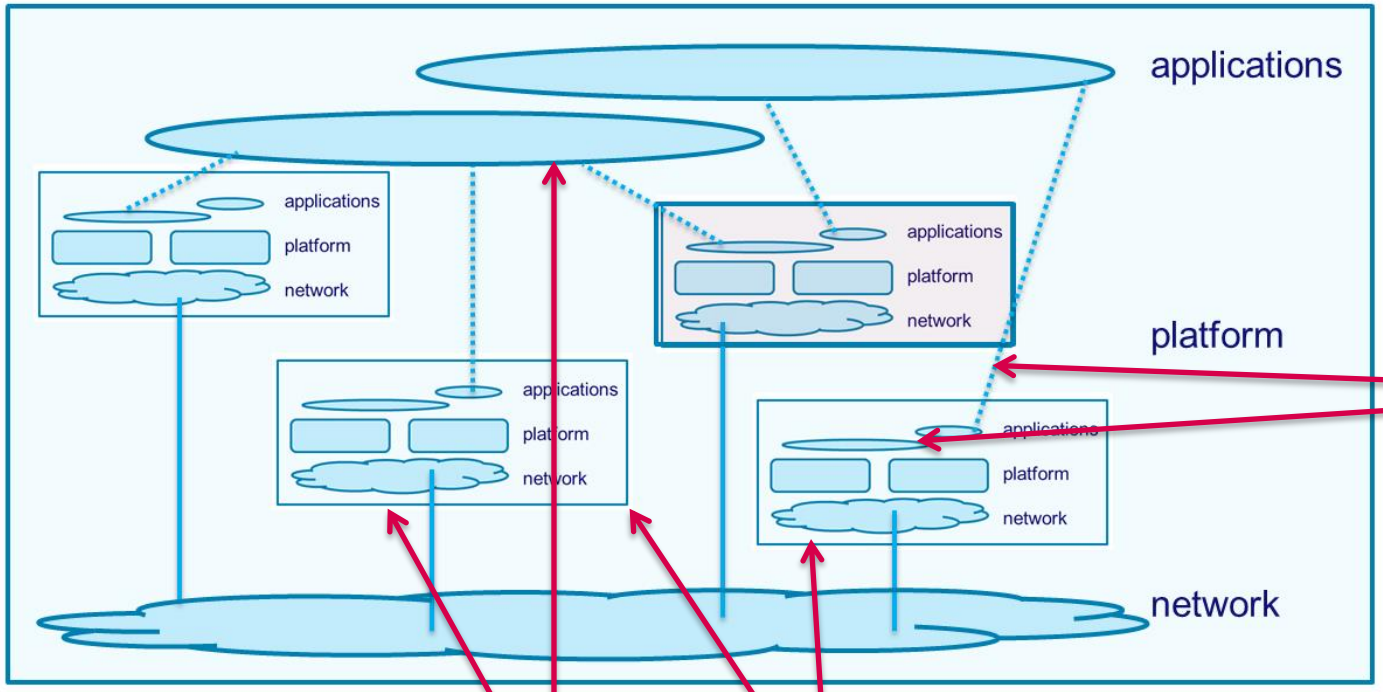


SoS characteristics

- *Operational Independence*
  - autonomous behavior and goal of subsystems
- *Managerial Independence*
  - subsystems managed by different authorities
- *Evolutionary Independence*

- *Geographic distribution*
- *Emergent behavior*
  - properties deriving from the combination of subsystems
  - properties difficult or impossible to deduce from subsystems
- *Heterogeneity*
- *Networks as integration point*

Composition of systems into SoS



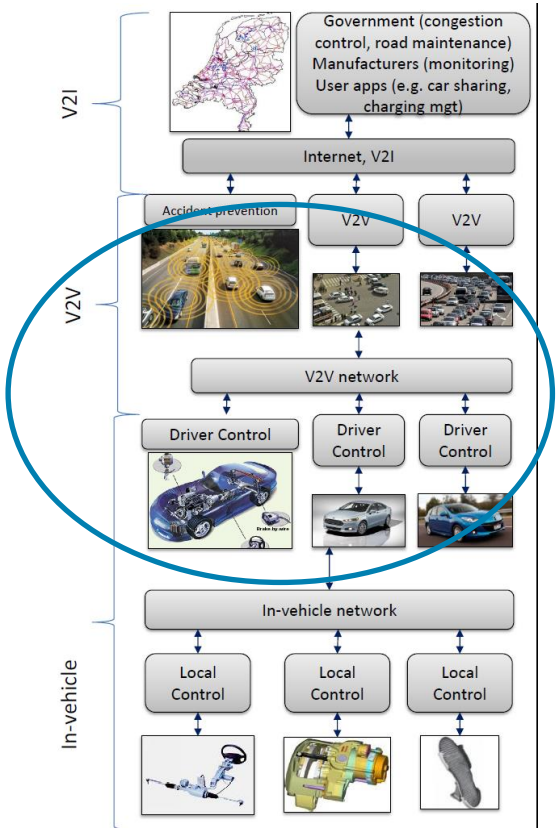
competition of control
- local versus global

uncorrelated requirements

architectural diversity

Two worlds

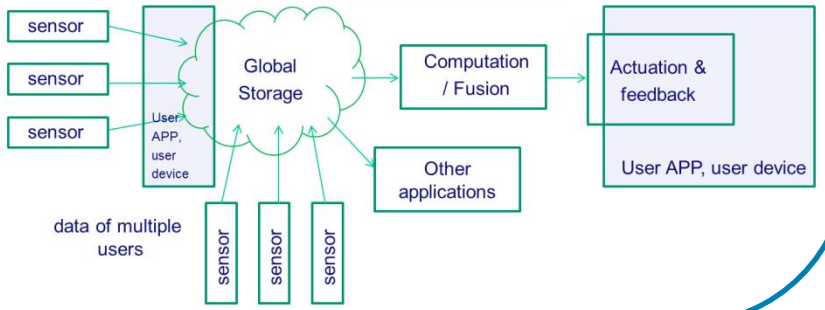
- Pervasive connectivity moves into the safety critical domain
  - by including actuation
  - by penetrating safety critical systems
  - *uncertainty* and concerns of *connectivity* and *scalability* are complemented with *timeliness* and *dependability*
- Safety critical systems (CPS) are becoming connected
  - by including open networks
  - *robustness* and (timing) *guarantees* are complemented with *scalability* and *uncertainty*



Connected domains

- Applications run on top of connected (managerial) domains
  - an application takes resources from that domain and possibly runs code
  - within a domain a single behavioral policy may be assumed
  - the overall application is emergent
- The perimeter is not that clear
  - the domain can be more logical than physical

- IoT *example: smartphone apps*
  - take resources from phone, network, back office cloud
  - crossing of managerial domains by user consent
    - sometimes policy, sometimes just black & white



How to engineer (compose) SoS applications?

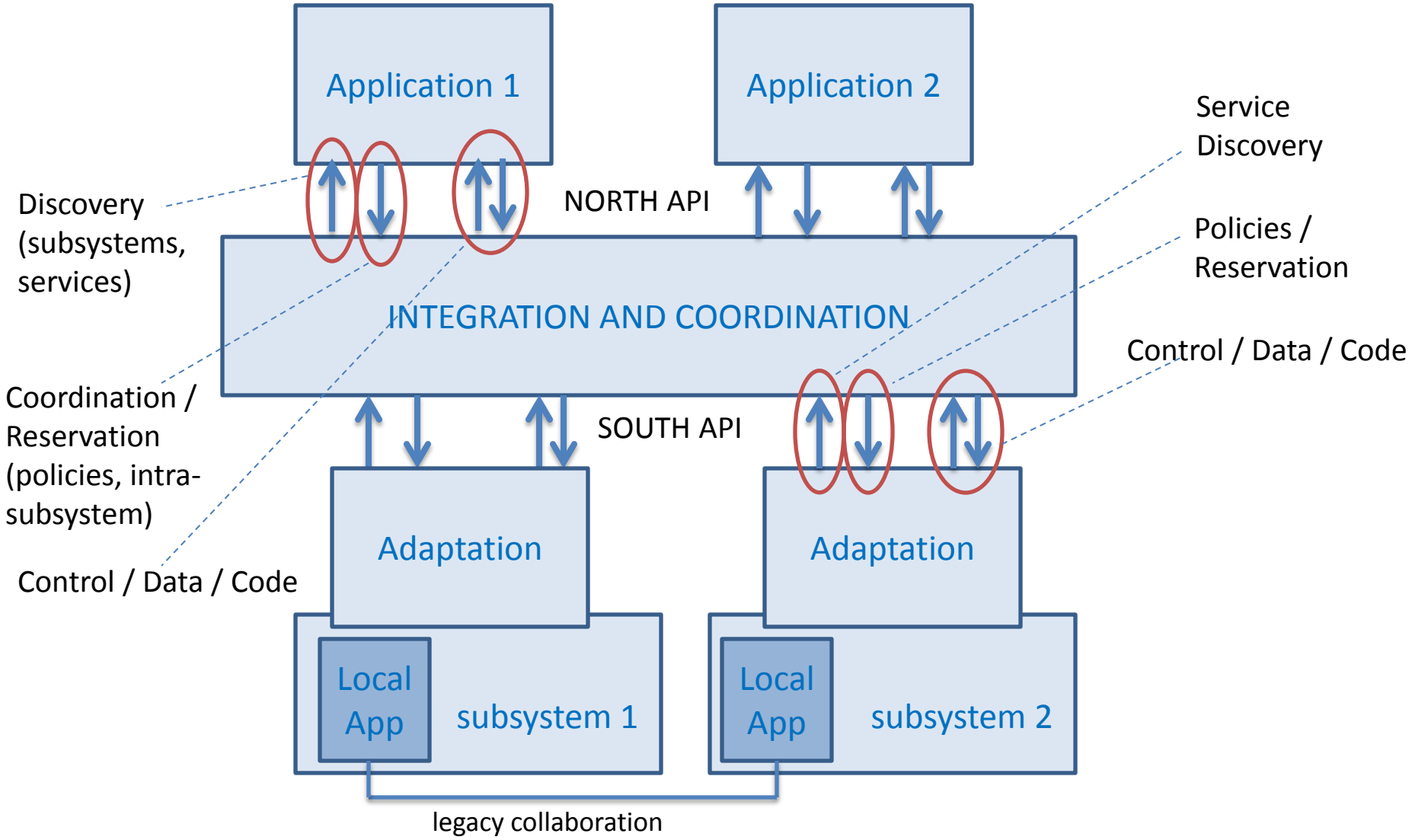
- E.g. a Smart City application
- Lewis et al.: three step life cycle
  1. list all software services in the concerned subsystems
  2. build the integrated SoS application from these
  3. examine and realize the consequences for the original subsystems
     - e.g. access to data, using computation and data resources

- Some thing like this has to be done *but*
  - this assumes a stable software ‘base’ (independent evolution!)
  - it requires subsystem managers to be involved
  - it invites adjustments of subsystems to the applications at hand leading to maintenance problems
    - …. reducing dependencies is key
  - it is difficult to involve third party application developers

Composition architecture – some requirements

- Separate the following
  - *development API* for application developers (“North side”)
  - *integration API* for subsystems (“South side”)
  - adaptation layer for subsystems

- Interfaces include
  - *reservation* of resources
  - *policies*, and negotiation thereof
  - *coordination* of SoS applications


Application 1
Application 2
Service Discovery
Discovery (subsystems, services)
NORTH API
Policies / Reservation
INTEGRATION AND COORDINATION
Control / Data / Code
Coordination / Reservation (policies, intra-subsystem)
SOUTH API
Adaptation
Adaptation
Control / Data / Code
Local App
subsystem 1
Local App
subsystem 2
legacy collaboration

In summary

- Pervasive networking and Safety Critical systems move closer and get mixed

- Both domains have characteristics of Systems of systems
  - composition of systems with an independent goal
  - composition of applications on top

- Such compositions should focus on:
  - reservations, extra-functional properties at interfaces
  - an explicit role for third party developers
  - avoiding increasing complexity in individual subsystems ad-hoc solutions