ERAF
EIROPAS REĢIONĀLĀS ATTĪSTĪBAS FONDS

IEGULDĪJUMS TAVĀ NĀKOTNĒ

EIROPAS SAVIENĪBA

INSTITUTE OF ELECTRONICS AND COMPUTER SCIENCE

A distributed data processing architecture for real time intelligent transport systems

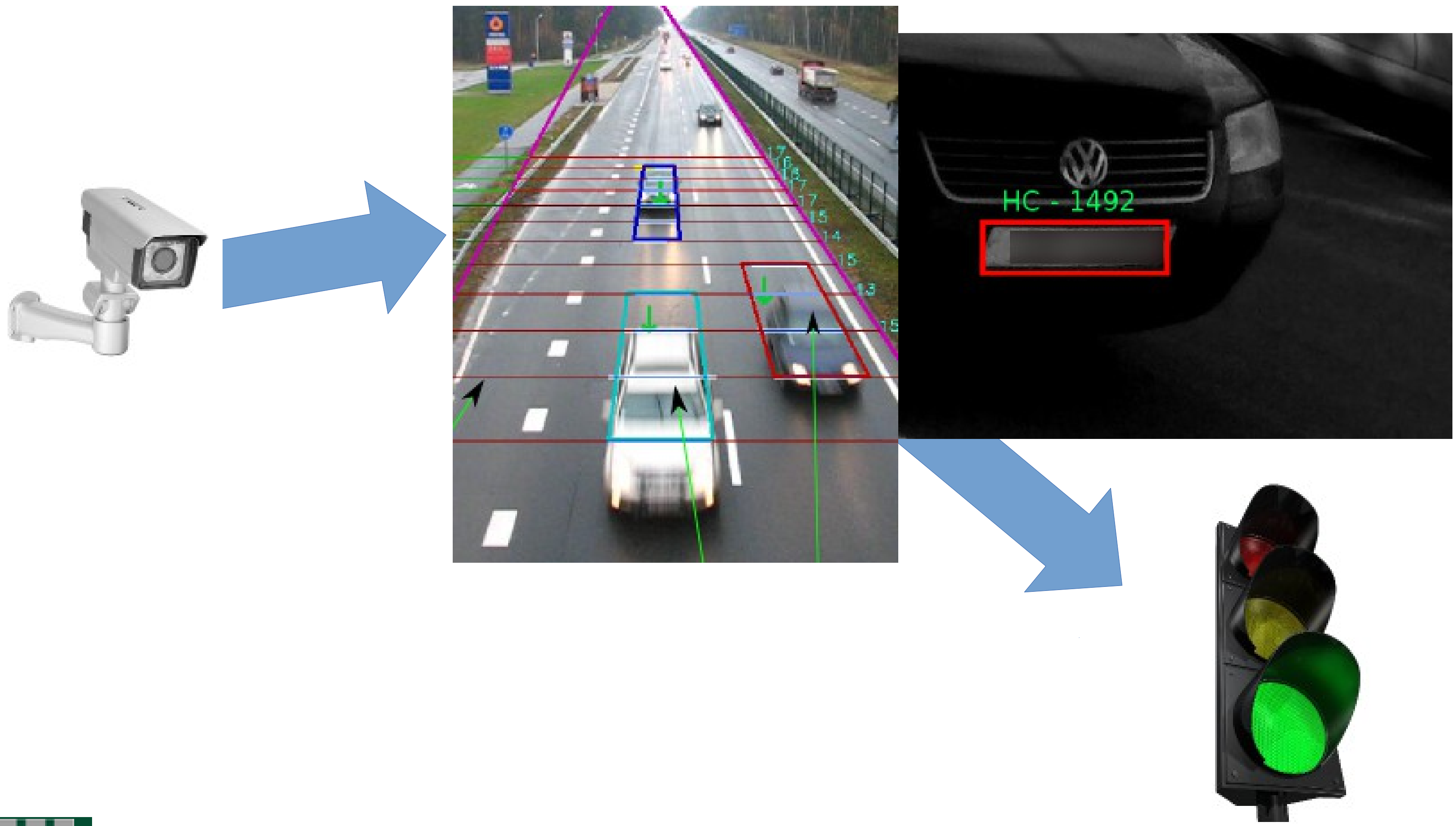
<u>K. Nesenbergs</u> (*krisjanis.nesenbergs@edi.lv*)
L. Selavo (leo.*selavo@edi.lv*)

Institute of Electronics and Computer Science
14 Dzerbenes street,
Riga, Latvia

This presentation is supported by project No.
2010/0250/2DP/2.1.1.1.0/10/APIA/VIAA/086

Goal

Real-time intelligent transport system



This specific paper

System architecture for a real-time intelligent transport system

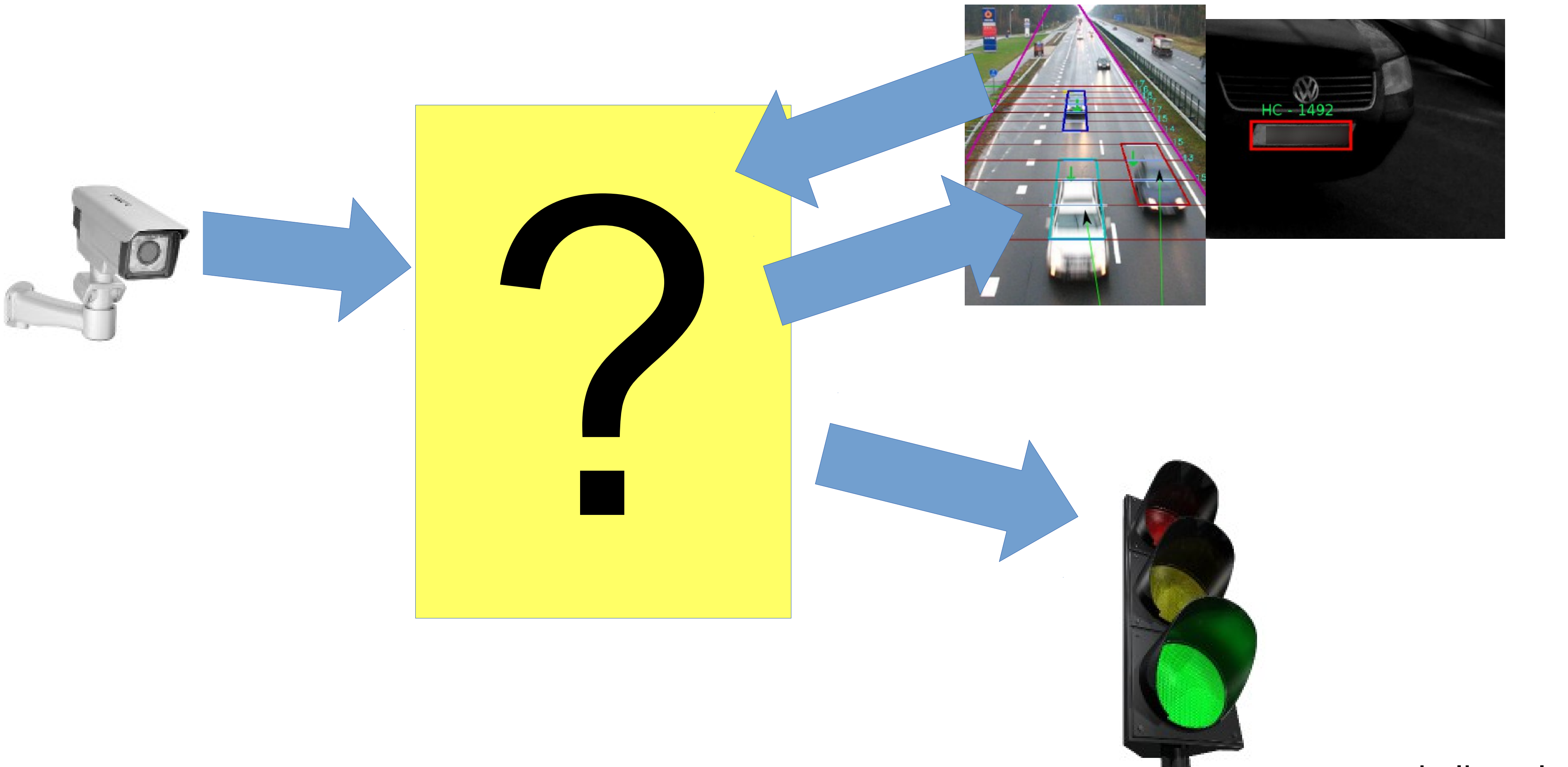


Figure from: www.skolioze.lv

Requirements

- Real-time functonality in life critical situations
- Facilitates easy development of data gathering, processing and decision making modules
- Redundant and reliable

Real time, life critical

- Distributed data processing with load balancing
- Fall back to local safety measures if system fails

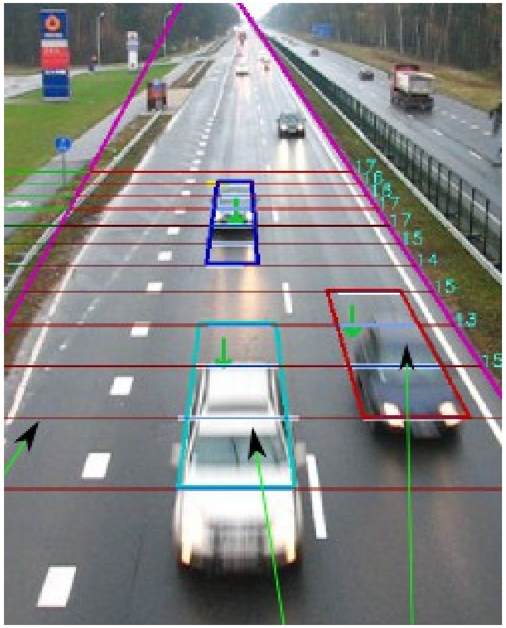
Redundancy, reliablility

- Data replication across several physical systems
- Capability to continue work even if several systems fail
- Resume with previous settings and data on restart
- Watchdog functionality

Modules

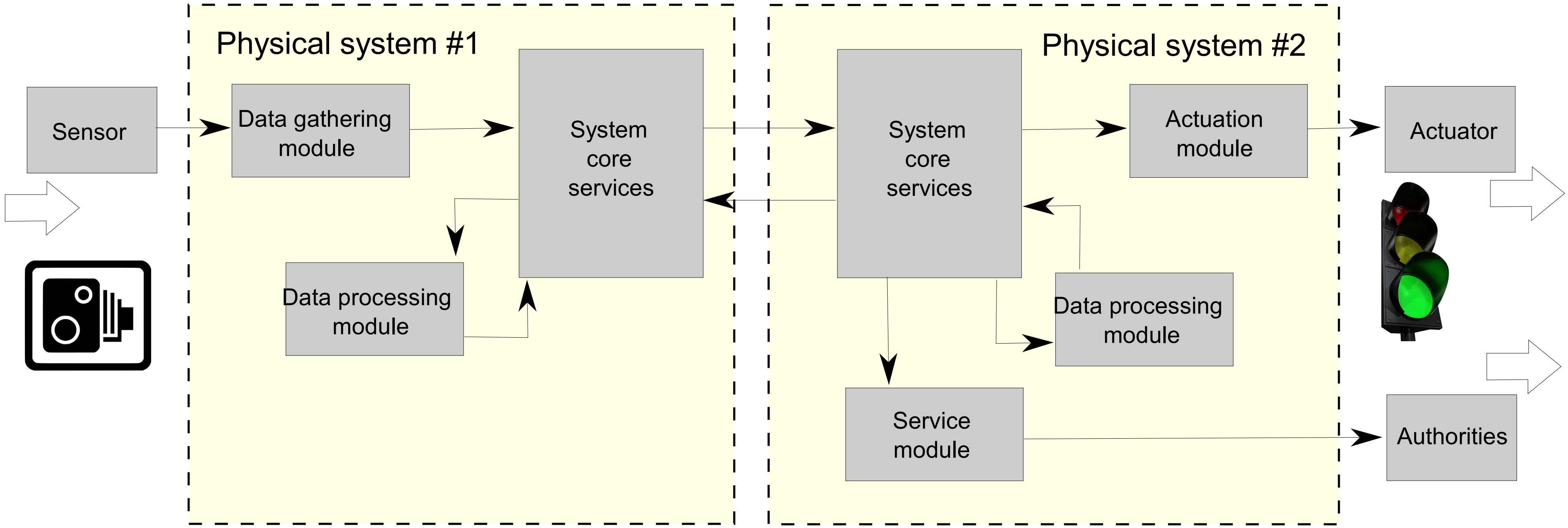
- Distributed system abstracted away from module developers
- Each module requests data from system as input, processes it independently and provides data as output
- Types:
  - Data gathering from sensors
  - Actuating external actuators
  - Data processing
  - Service (logging etc.)

Module examples:



- Vehicle detection in video
- Number plater location in video
- Car detection using microphone arrays, magnetic sensors and other sensors
- Merging data for complex vehicle detection
- Calling emergency services
- Controlling traffic lights

Architecture



System core services

- Configuration subsystem
- Data storage subsystem
- Inter-core network subsystem
- Watchdog subsystem
- Data replication subsystem
- Data retrieval subsystem

Configuration subsystem

- Keeps configuration consistent across physical systems and system restarts
- Distributes external configuration changes for the active and further sessions

Data storage subsystem

- Stores all relevant tracking data on each physical system seperately
- Initially modules send data to local core only
- Data distribution controlled by data replication subsystem

Inter-core network subsystem

- Message distribution and routing between physical systems
- System routing information
- System load information

Watchdog subsystem

- Watches heart-beat signals from all cores
- Takes necessary action to restart failed system parts and/or sends notifications to appropriate personnell

Data replication subsystem

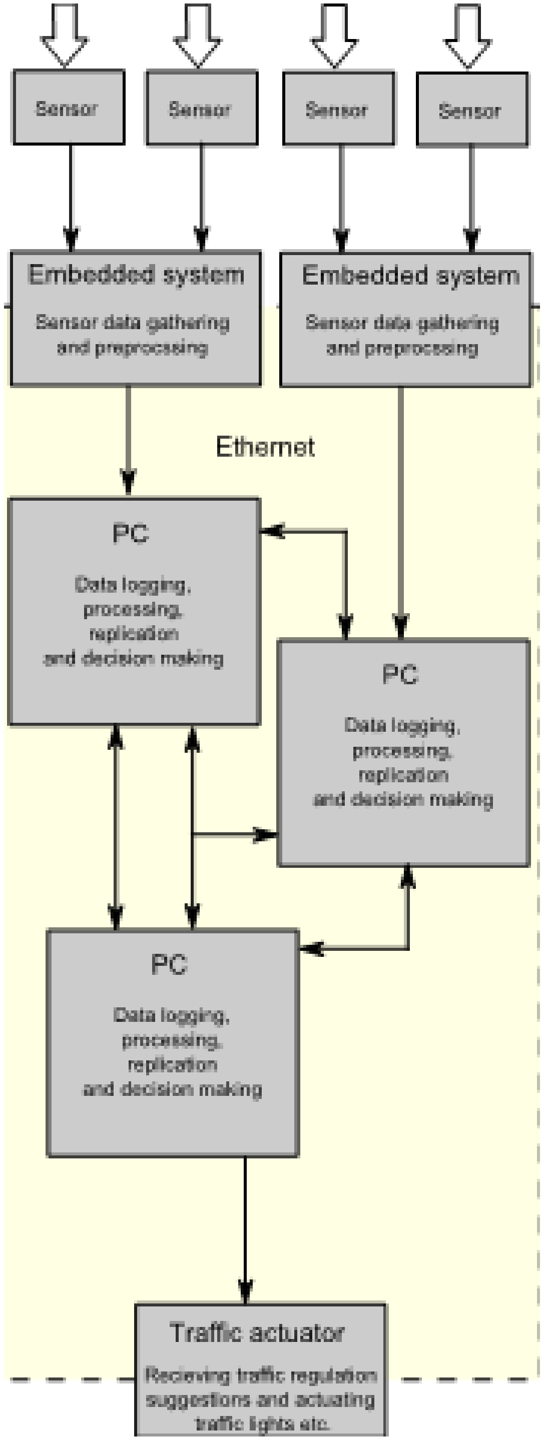
- All data packets are repliacted to at least N physical systems
- After recieving data packet each core determines if it should be stored and/or sent for further replication or dropped completely
- Each packet contains information of cores containing it.
- Replication targets determined by reported system load
- If cores containing some data fail, watchdog initiates additonal replication of the involved data

Data retrieval subsystem

- Modules request data from local core
- Local core determines which of system cores contains the freshest data of type and source requested and connects requesting module to the appropriate core
- Either one packet or data stream can be requested

Physical testbed implementation

- 3 Mini-ITX personal computers (i5, 8GB RAM)
- 8 Raspberry PI embedded computers
- Wireless router
- Sensors (cameras, magntic sensors, microphone arrays etc.)
- Traffic actuators (simulated, because of birocratic reasons)



Conclusions

- A distributed data processing architecture proposed and implemented in test environment
- Redundancy and fault tolerance allows system to continue working even with several physical systems down.
- Load distribution allows additional benefits, such as capability to run identical high intensity data processing modules in parallel on several systems
- Module developers do not have to concern them selves about the distributed system architecture reducing module development time and complexity
- At the moment only simulated data on real physical system, will be tested in real traffic environments

Thank you for your attention!