Center for Embedded Systems

An NSF Industry/University Cooperative Research Center

Distributed run-time management for multi-agent system

Iraklis Anagnostopoulos,
Southern Illinois University

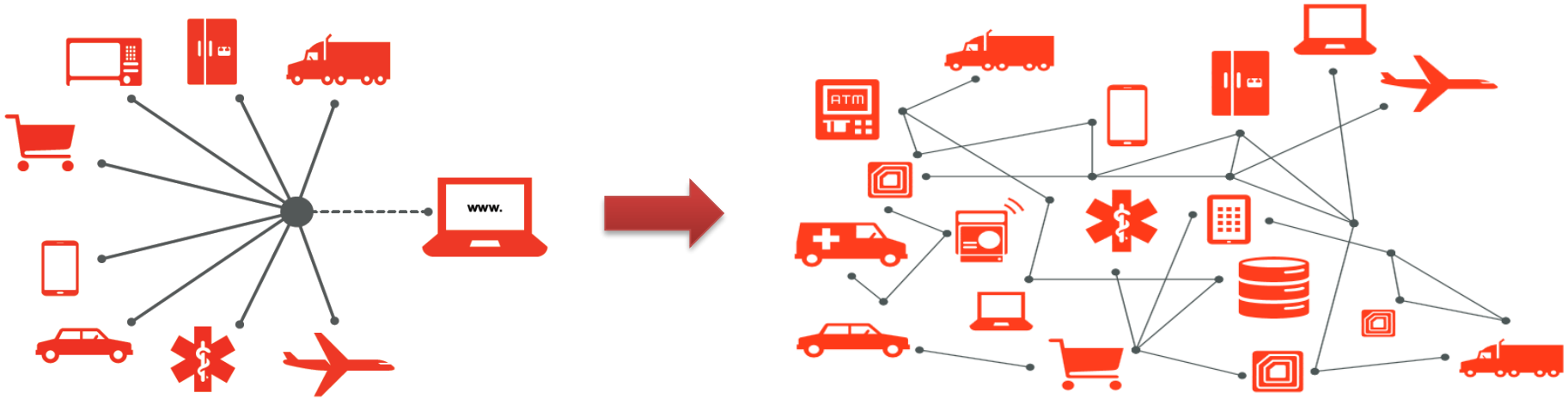
SIU Southern Illinois University CARBONDALE

National Science Foundation

ASU Ira A. Fulton Schools of Engineering
ARIZONA STATE UNIVERSITY

Project Overview and Description

- By 2020 16B dynamic-networked devices will be deployed
  - Increased complexity
  - Increase error probability
  - Node management and connectivity
- Necessity for **distributed run-time management**



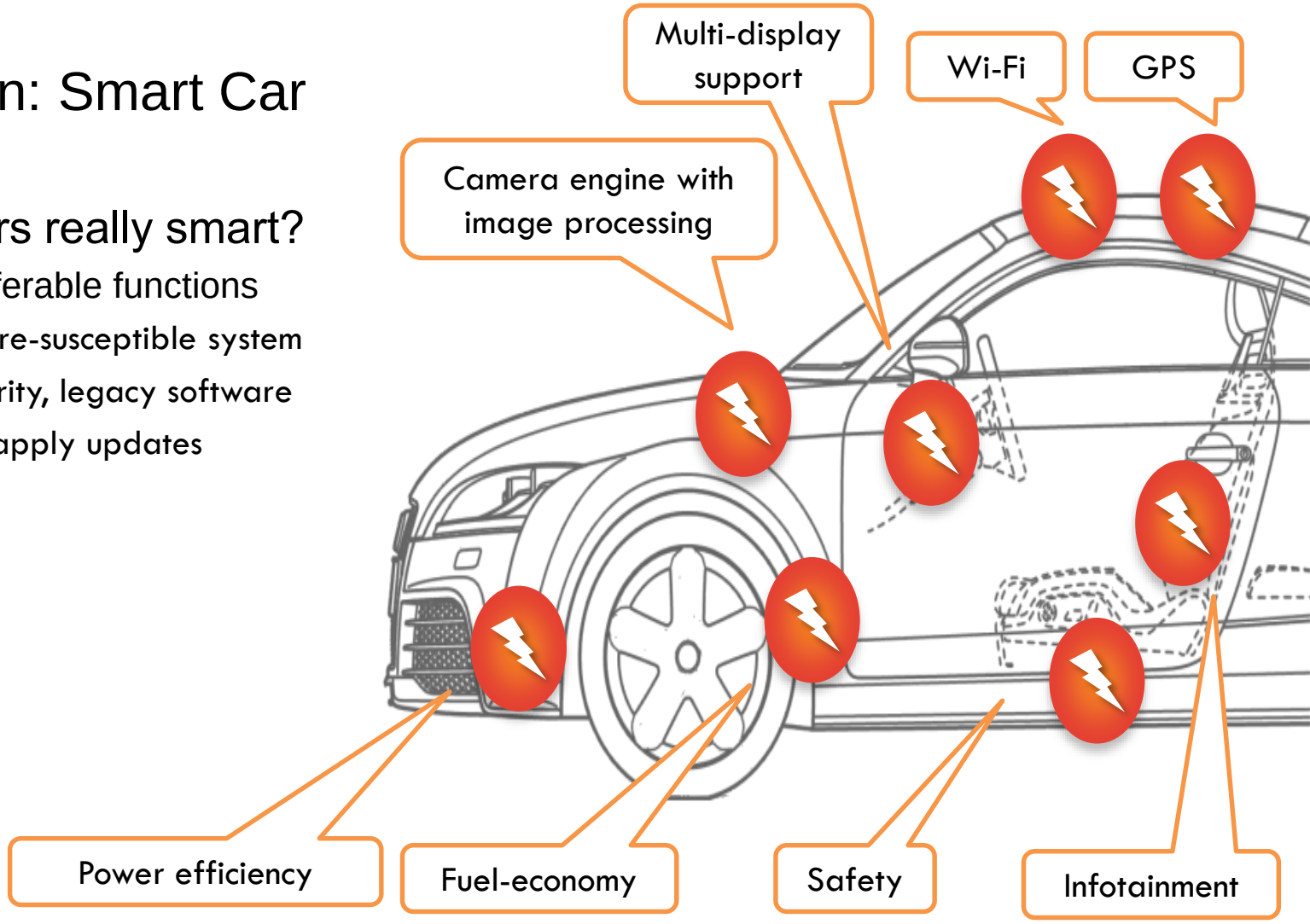
Source: Oracle

Project Overview and Description

- Application: Smart Car

Are “Smart” cars really smart?

- Non-transferable functions
- Highly failure-susceptible system
- Aging, security, legacy software
- Difficult to apply updates



Source: ITU Workshop

Approach

- Propose a **distributed framework for run-time management** of multi-agent systems
  - Distributed agents
  - Self-optimization
  - Node discovery
  - Resource mapping by hypervisors

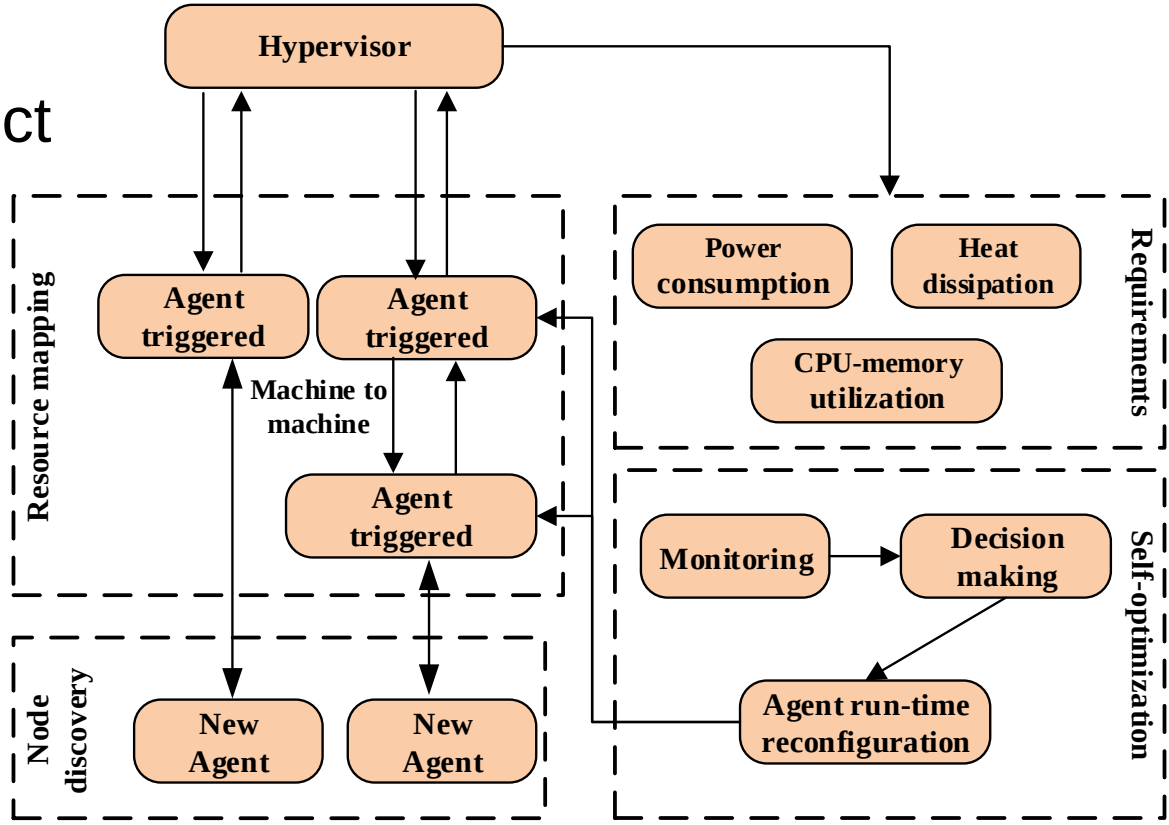
- Novelty and benefits:
  - Couple run-time services in a distributed way
  - Respect system requirements
  - Self-managed system functionality
  - the concept of multi-agent systems in modern automotive environment will be integrated and tested

Project Tasks/ Deliverables

	Description	Date
1	Study of existing techniques and selection of the appropriate methodologies-algorithms	12/2015
2	Define architecture and hardware specifications for hypervisor and agents	4/2016
3	Comprehensive report about the implemented algorithms and techniques for distributed multi-agent systems	8/2016
4	Demo with interconnected physical boards (e.g. beagleboard, raspberry pi over ethernet) of a simple scenario	8/2016

Executive Summary

- **Distributed run-time management**
  - Hypervisors
  - Requirements respect
  - Resource mapping
  - Node discovery
  - Self-optimization



REMAINING SLIDES For poster session ONLY (as many as you'd like)

TECHNICAL DETAIL

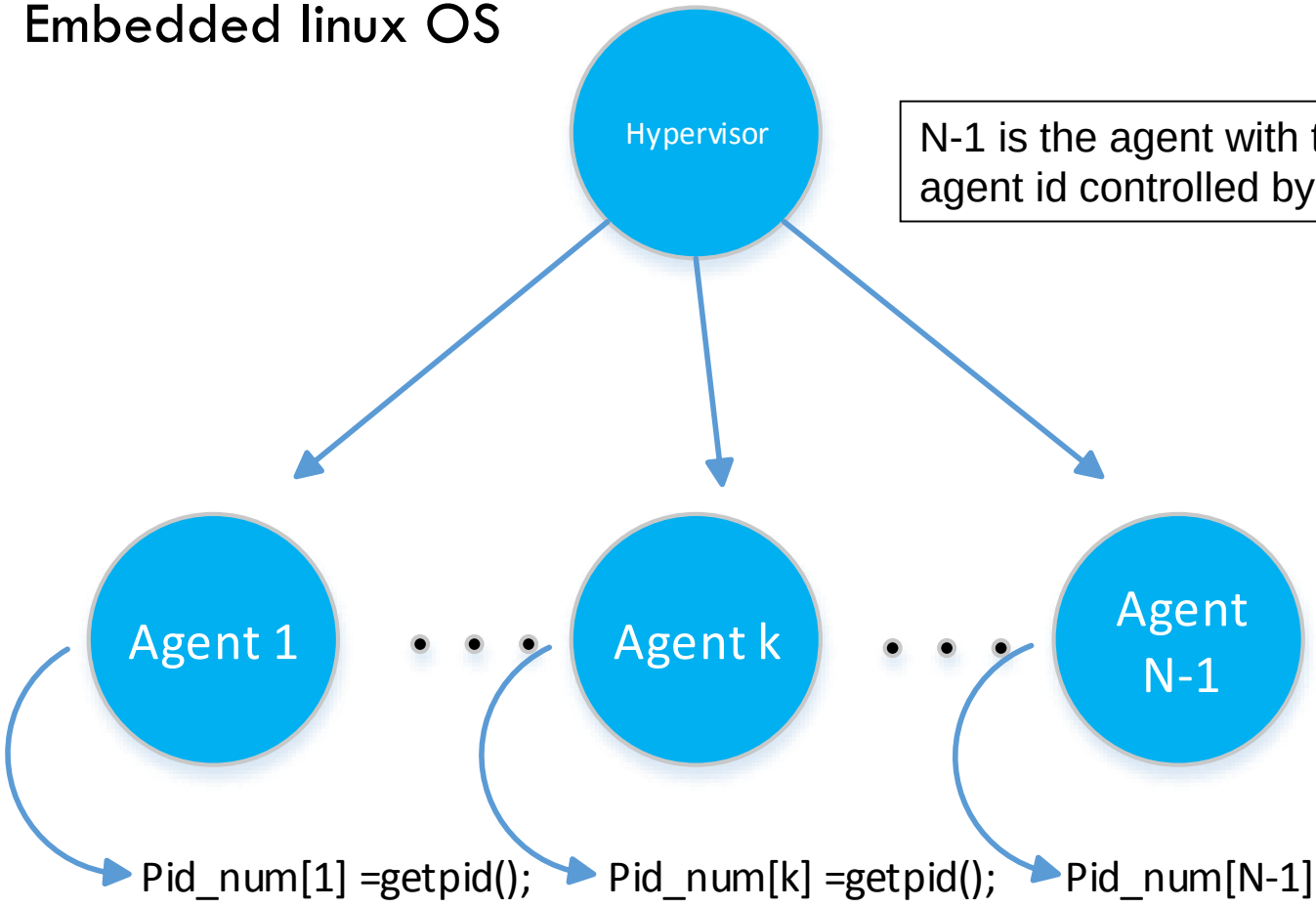
ECU classification

- Hypervisors
  - Handles all the unoccupied ECUs
  - Checks some of system's requirements
  - Communicates with agents
  - Monitoring
  - They are not the center of communication

- Agents
  - Task/Application execution
  - Self-optimization process
  - Respect system requirements
  - Self-managed system functionality

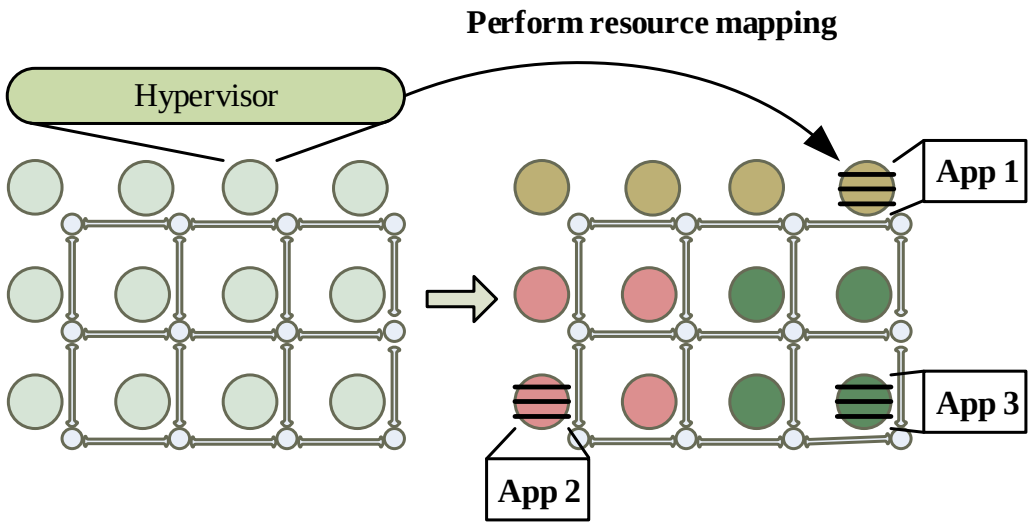
Run-time management

- RT Linux kernel
  - Deploy on raspberry pi
  - Embedded linux OS



Distributed run-time management

- Divide& Conquer



Anagnostopoulos, I.; Bartzas, A.; Kathareios, G.; Soudris, D., "A divide and conquer based distributed run-time mapping methodology for many-core platforms," Design, Automation & Test in Europe Conference & Exhibition (DATE), 2012 , vol., no., pp.111,116, 12-16 March 2012

- Monitoring
  - Code instrumentation
  - Local statistics → Workload prediction
  - Decision making