# Cloud computing 2018
# Panel on CLOUD/SERVICES

## Challenges in Cloud Computing-based Systems

**Moderator**

**Yong Woo LEE, Ph.D.**

**Professor, University of Seoul**

**President, Smart Consortium for Seoul, Korea**

**Chair, Seoul Grid Center**

**Chair, The Korean National Committee for ISO JTC1/SC22**

## 2018. 2. 21, Barcelona

## ● Moderator

Yong Woo Lee, University of Seoul, Korea

## ● Panelists

- Yong Woo Lee, The University of Seoul, South Korea
- Kerry Long, Program Manager, IARPA, USA
- Eugen Borcoci, University Politehnica of Bucharest, Romania
- Raimund Ege, Northern Illinois University, USA
- Nane Kratzke, Lübeck University of Applied Sciences, Germany

# Challenges in Cloud Computing-based Systems

1. System side

2. Application side : Cloud-native applications (CNA)

3. Hybrid.

# Challenges in Cloud Computing-based Systems

1. System side

* QoS/SLA in cloud computing
  - Real-time cloud computing
* Security
  - Vulnerability in cloud computing
  - Safety and privacy in cloud computing
* Migration
* Edge/Fog computing
* Standardization in cloud computing.

# Challenges in Cloud Computing-based Systems

2. Application side : Cloud-Native Applications (CNA)

- Cloud computing for big data processing
- Cloud computing for streamed data processing
- Cloud computing for stream reasoning
- Cloud computing for smart cities
- Cloud computing for smart buildings
- Cloud computing for IoT
- Cloud computing for smart systems.
- Etc.

# Discussion

# Conclusion


תודה
Dankie
Gracias
Спасибо
شكراً
Merci
Takk
Köszönjük
Terima kasih
Grazie
Dziękujemy
Dėkojame
Ďakujeme
Vielen Dank
Paldies
Kiitos
Täname teid
谢谢
Thank You
Tak
感謝您
Obrigado
Teşekkür Ederiz
감사합니다
Σας Ευχαριστούμ
ขอบคุณ
Bedankt
Děkujeme vám
ありがとうございます
Tack

# Panel on Cloud Computing

## Topic: Challenges in Cloud Computing-based Systems

## Edge-oriented computing and network slicing

**Eugen Borcoci**

**University POLITEHNICA Bucharest**
**Electronics, Telecommunications and Information Technology Faculty ( ETTI)**
Eugen.Borcoci@elcom.pub.ro

**ComputationWorld 2018, February 17-22, Barcelona**

## Why network slicing?

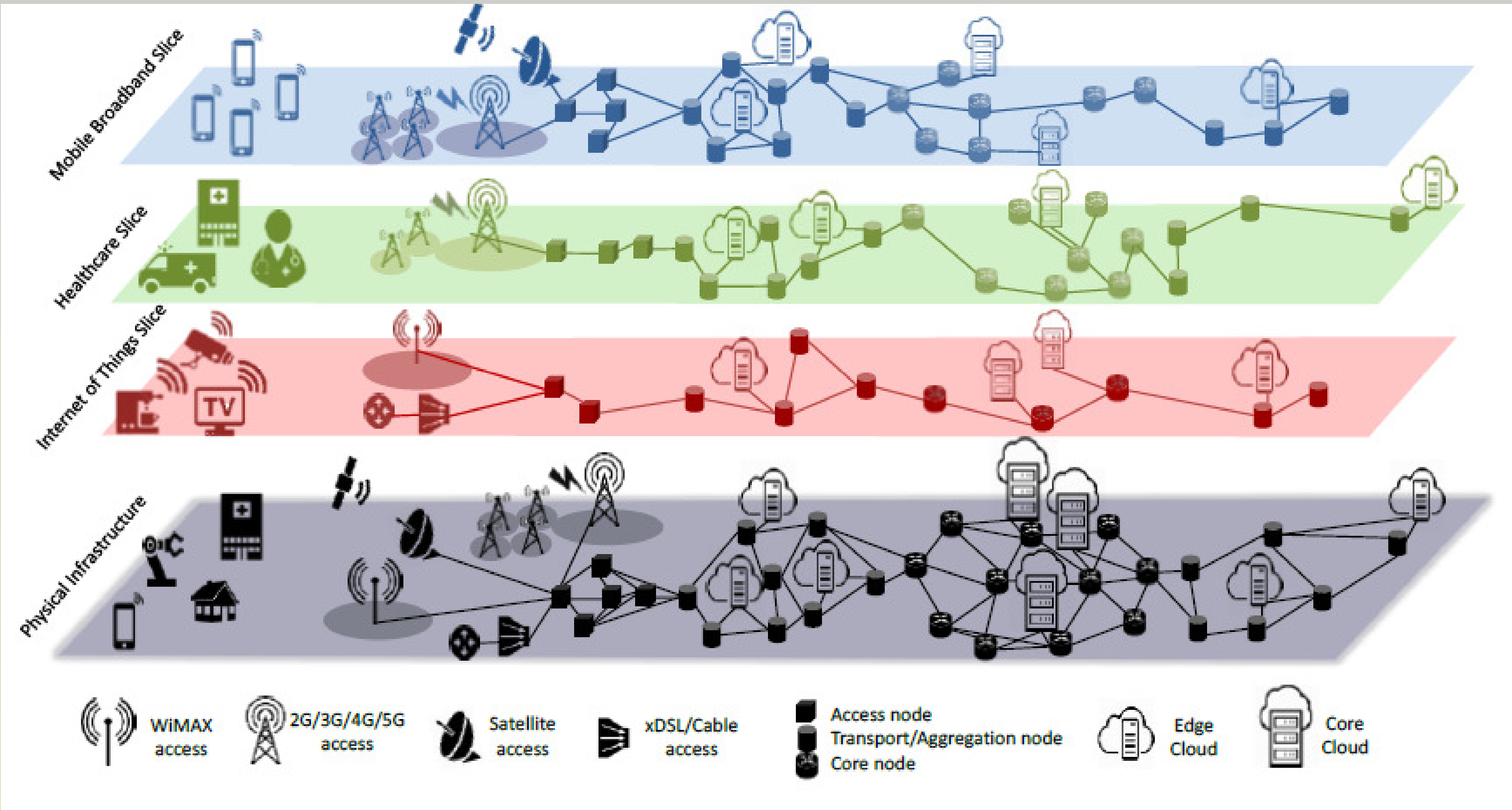
- **Network softwarization** - an emerging trend – **to transform the networks using software-based solutions**
  - **multiple logical (virtual) networks (*network slices*)** each tailored for a given class of use/business cases, on top of a shared physical network.
- **Network slices :**
  - **end-to-end (E2E) logical networks**
  - **running on a shared underlying (physical or virtual) network**
  - **mutually isolated**
  - **with independent (customisable) control and management**
  - **created on demand (short/ medium/ long life cycle)**
- **Network slice:** virtual network, flexible enough to simultaneously accommodate diverse business-driven use cases requirements from multiple players on a common network infrastructure

# Edge-oriented computing and network slicing

- **Why network slicing?** (cont'd)
- **Driving factors**
  - **Users - services and applications view**
    - numerous novel applications and services
      - IoT in smart cities, industry, economics, transportation, health, social entities, etc.
    - Tenant independency to configure and manage its slice
    - On demand services
    - Cloud computing facilities available
    - **…..**
  - **Operators' needs**
    - Flexibility
    - Programmability
    - Efficient usage of resources
    - More powerful network and services management
    - Isolation of classes of services and applications
    - E2E QoS/QoE controllable degree of guarantees, SLAs, …
    - Better security
    - **…..**
- ***Where to apply slicing?***
  - ***5G, Core, -spanning multi-domain, E2E***

# Edge-oriented computing and network slicing

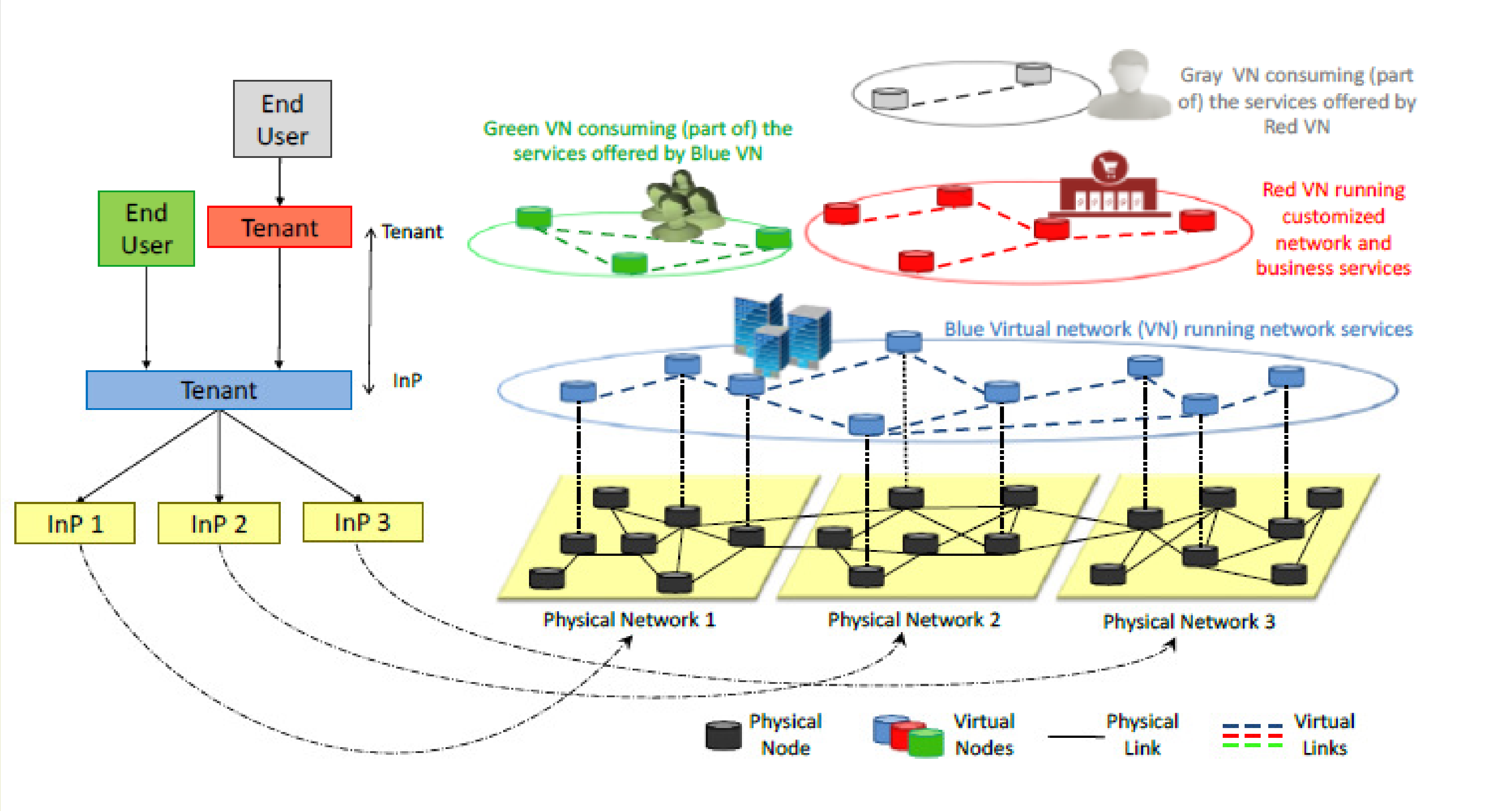
- **Network slicing - example**




*Source: J. Ordonez-Lucena et. al., , Network Slicing for 5G with SDN/NFV: Concepts, Architectures and Challenges " IEEE Comm. Magazine, 2017*

# Edge-oriented computing and network slicing

- **Network slices: recursive model**




*Source: J. Ordonez-Lucena et. al., , Network Slicing for 5G with SDN/NFV: Concepts, Architectures and Challenges " IEEE Comm. Magazine, 2017*

# Edge-oriented computing and network slicing

## Cloud Computing - evolution

- **Cloud computing (CC) services -**initially offered by data centers (including public/private/community clouds) – intensively used
- **Centralisation** (processing and storage) in traditional CC -> **limitations**
- **Services and apps. like IoT, mobility-related, M2M, ..**
  - **requirements:** Low latency/response time, high bandwidth, location and context awareness, reduction in amount of data transferred to CC and back
- **Solution**
  - *Edge-oriented computing – cooperating with CC*
    - *Multi-access/Mobile Edge Computing (MEC)*
    - *Fog Computing (FC)*
    - *Cloudlets*
    - *Micro-Data Centers*
    - *.....*
- **Auxiliary technologies**
  - *Virtualisation techniques, Software Defined Networks (SDN)*
  - *Network Function Virtualisation (NFV- ETSI)*

# Edge-oriented computing and network slicing

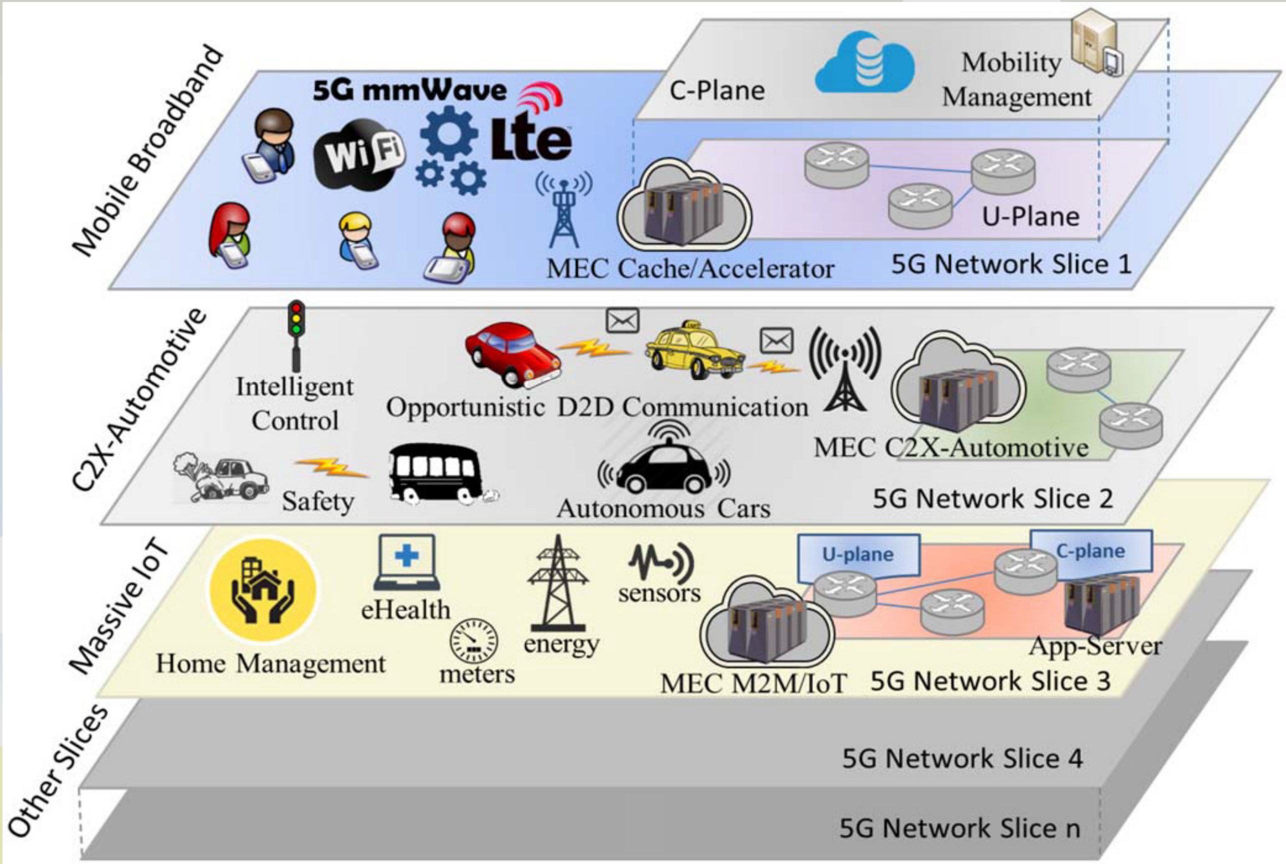
- **Open research domain**
  - **How to create network slices- in a distributed cloud/edge computing environment ?**

# Edge-oriented computing and network slicing

- **How to create network slices- in a distributed cloud/edge computing environment ?**
  - **Distributed (edge-oriented) cloud infrastructure** - candidate to be the **backbone of the 5G network**
    - It is a good environment for **decomposing the large monolithic network** functions of legacy HW into a catalog of modular network capabilities/functions of varying granularity
    - The capabilities can be flexibly chained (applying NFV style) to form network slices to support the diverse services
- 
  - The network **slice design** can be **continuously optimised** based on collected information at run-time from the network and services
- 
  - **Autonomic network management functions, analytics, machine learning**, could enable the automation and optimisation of the network slice life cycle (design, creation, monitoring, optimization and deletion)

# Edge-oriented computing and network slicing

- **Example: Mobile Edge Computing (MEC)- possible role in 5G slicing**




*Source: T.Taleb, et al., "On Multi-Access Edge Computing: A Survey of the Emerging 5G Network Edge Cloud Architecture and Orchestration" IEEE COMMUNICATIONS SURVEYS & TUTORIALS, VOL. 19, NO. 3, THIRD QUARTER 2017 1657*

# Edge-oriented computing and network slicing

## Conclusions

- **5G slicing and cloud/edge computing technologies – can be used in a combined way**
  - promising solution for creation of an environment to support a large range of requirements for Future Internet services and applications
  - **Some research challenges**
    - Dynamic autonomic management and resource allocation
    - Orchestration of virtualised infrastructures
    - Multi-domain aspects
    - E2E, business models, SLAs
    - Slice Isolation and Security
    - Big data, analytics capabilities
    - Scalability
    - Real time aspect for critical applications
    - Harmonising the solutions with SDN, NFV, virtualisation technologies – including standard and open solutions
    - Backward compatibility
    - Costs/benefits??

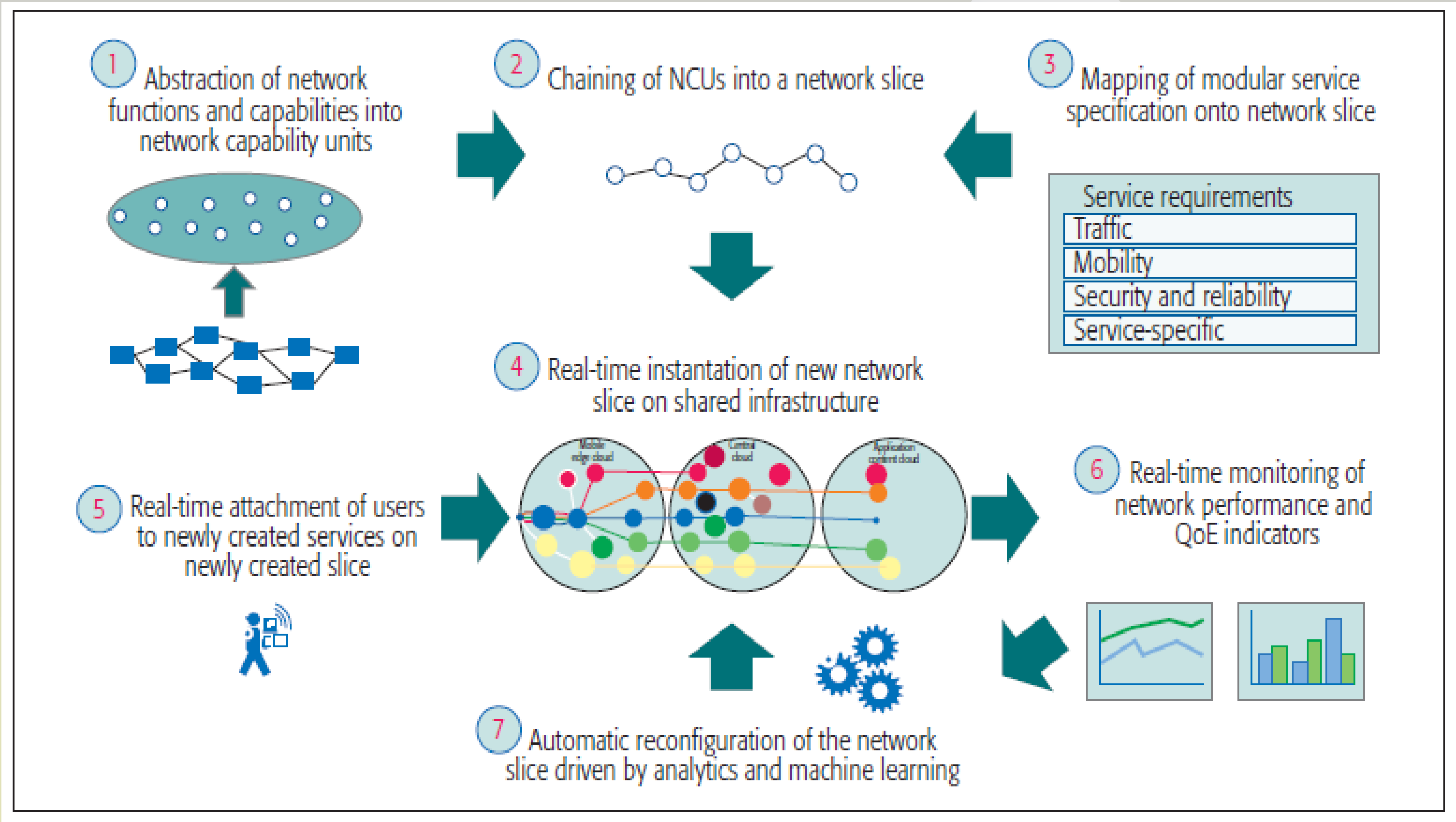
# Edge-oriented computing and network slicing

- **References**


1. J. Ordonez-Lucena et. al., , Network Slicing for 5G with SDN/NFV: Concepts, Architectures and Challenges " IEEE Comm. Magazine, 2017
2. Sameerkumar Sharma, Raymond Miller, and Andrea Francini, "A Cloud-Native Approach to 5G Network Slicing", IEEE Communications Magazine, August 2017
3. T.Taleb, et.al., "On Multi-Access Edge Computing: A Survey of the Emerging 5G Network Edge Cloud Architecture and Orchestration" IEEE COMMUNICATIONS SURVEYS & TUTORIALS, VOL. 19, NO. 3, THIRD QUARTER 2017 1657

# Edge-oriented computing and network slicing

## Life cycle of a cloud-native network slice




*Source: S.Sharma, R.Miller, and A.Francini, "A Cloud-Native Approach to 5G Network Slicing", IEEE Communications Magazine, August 2017*

*Panel Discussion: "Challenges in Cloud Computing-based Systems"*

# We have the Bricks to Build Cloud-native Cathedrals

***But do we have the mortar?***

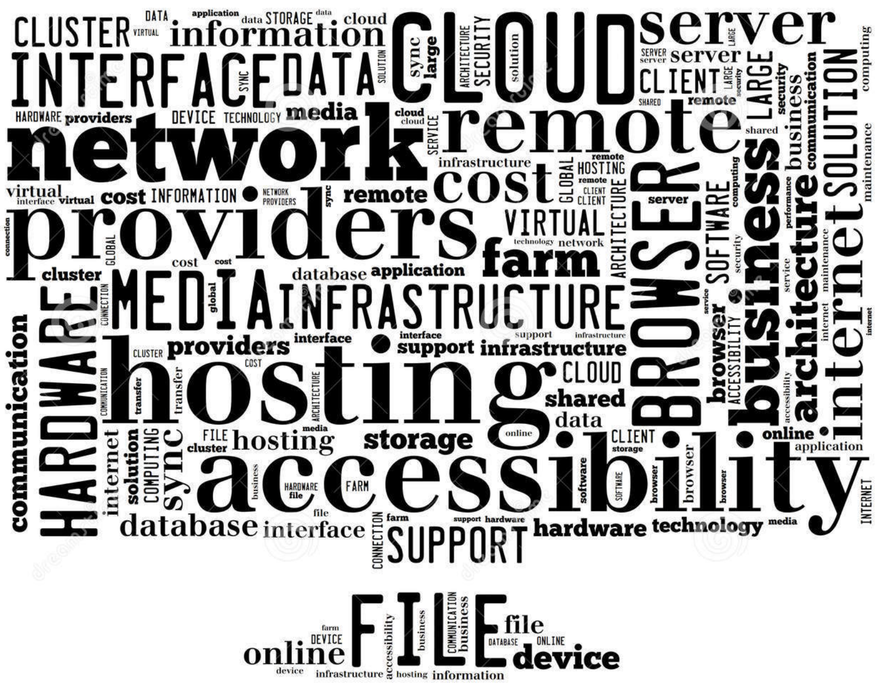
Nane Kratzke

FACH HOCHSCHULE LÜBECK
University of Applied Sciences

CoSA
Center of Excellence

# The Bricks of Cloud-native Systems






- IaaS
- PaaS
- SaaS
- FaaS + BaaS
- DBaaS
- CaaS
- …
- XaaS

# The Mortar of Cloud-native Systems

# Independent Systems Architecture (ISA)
## That is how practitioners build cloud applications

FACH HOCHSCHULE LÜBECK
University of Applied Sciences

1. *Modules (and Interfaces)*
2. *Separate Processes (Container)*
3. **Macro / Micro Architecture**
4. *Integration (limited and standardized)*
5. *Communication (limited set of protocols)*
6. *Independent continuous delivery pipeline (per module)*
7. *Standardized operation (across all modules)*
8. *Standards: enforced via interfaces*
9. *Resilience (dependent service failures)*

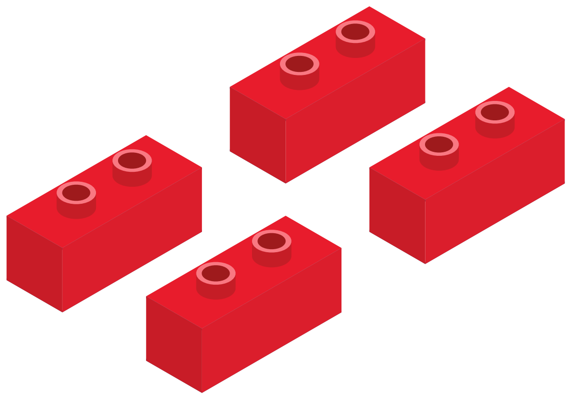
**PRINCIPLES**

Due to Infrastructure as Code even the Macro Architecture can be made executable.

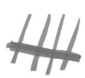
# Two Architecture Levels



## MACRO Architecture



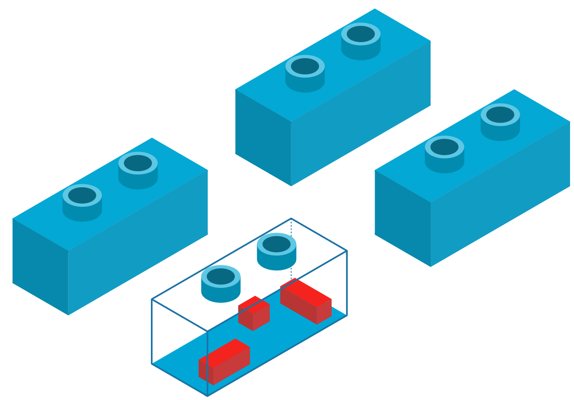
Decisions for all modules



Only very few languages

TOSCA, maybe Jolie

## MICRO Architecture



Decisions for one module

**Thousands** of languages

(each module can use its own)

## What programming language is currently covering this kind of complex integration and orchestration as its primary purpose?


- It took decades and plenty of languages to get only a handful of them for pragmatic application programming on single nodes.
- Shall we really trust the most dominant orchestration language that established for cloud infrastructure deployments and multi-host environments?
- Especially if we know, it has shortcomings regarding elasticity and transferability at runtime?

# Acknowledgement

FACH HOCHSCHULE LÜBECK
University of Applied Sciences

*Our research is funded by German Federal Ministry of Education and Research (13FH021PX4).*

FORSCHUNG AN FACHHOCHSCHULEN

Speaker Deck

Presentation URL

## Picture Reference

- **Ruine:** Pixabay (CC0 Public Domain)
- **Definition:** Pixabay (CC0 Public Domain)
- **Power Suply:** Pixabay (CC0 Public Domain)
- **Doors:** Pixabay (CC0 Public Domain)
- **Air Transport:** Pixabay (CC0 Public Domain, WikiImages)

# About

University of Applied Sciences

Nane Kratzke

**CoSA:** http://cosa.fh-luebeck.de/en/contact/people/n-kratzke

**ResearchGate:** https://www.researchgate.net/profile/Nane_Kratzke

**Blog:** http://www.nkode.io

**Twitter:** @NaneKratzke

**GooglePlus:** +NaneKratzke

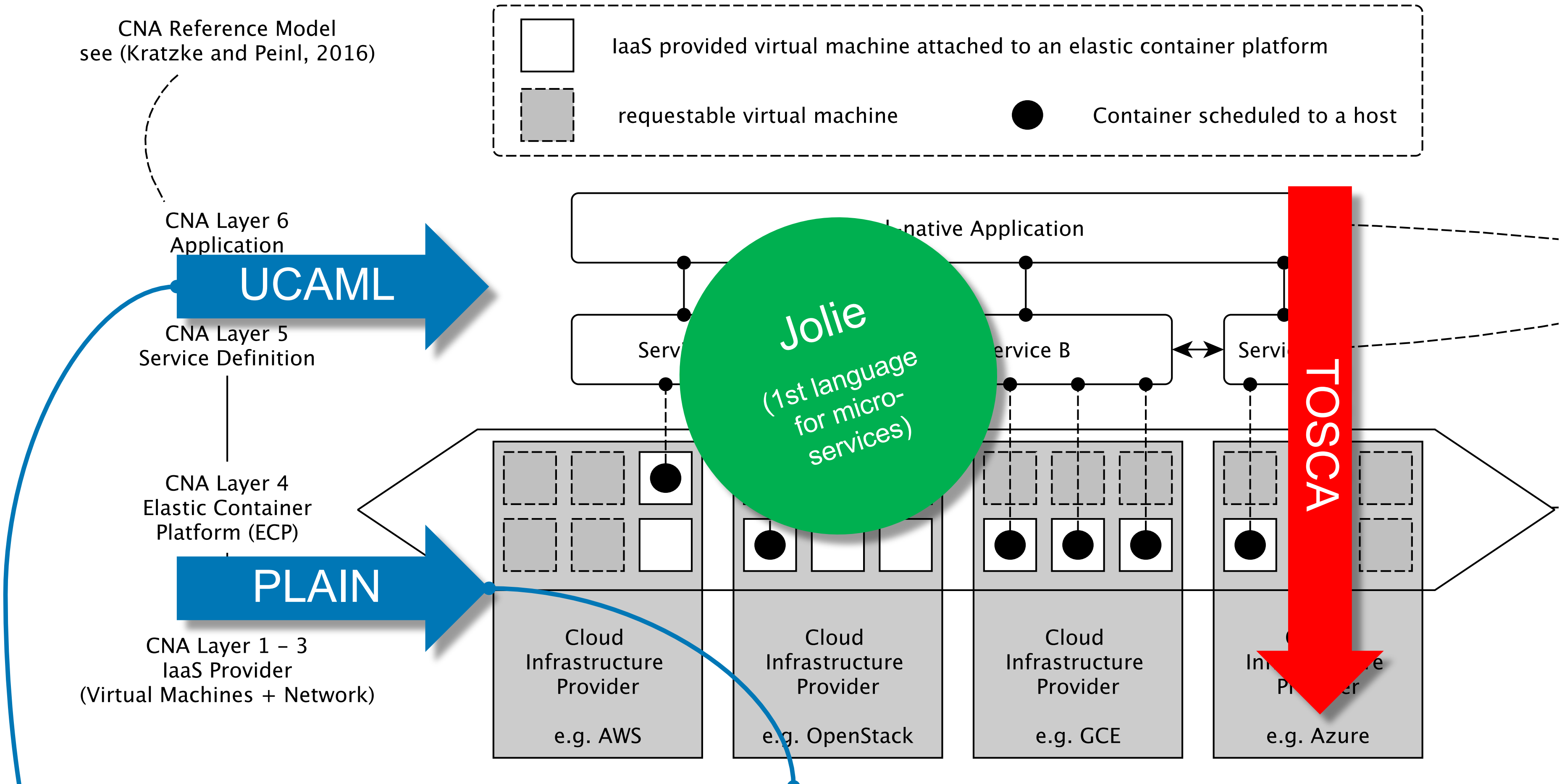
**LinkedIn:** https://de.linkedin.com/in/nanekratzke

**GitHub:** https://github.com/nkratzke

**SlideShare:** http://de.slideshare.net/i21aneka

# Backup Slides

# Can we solve cloud orchestration problems different?

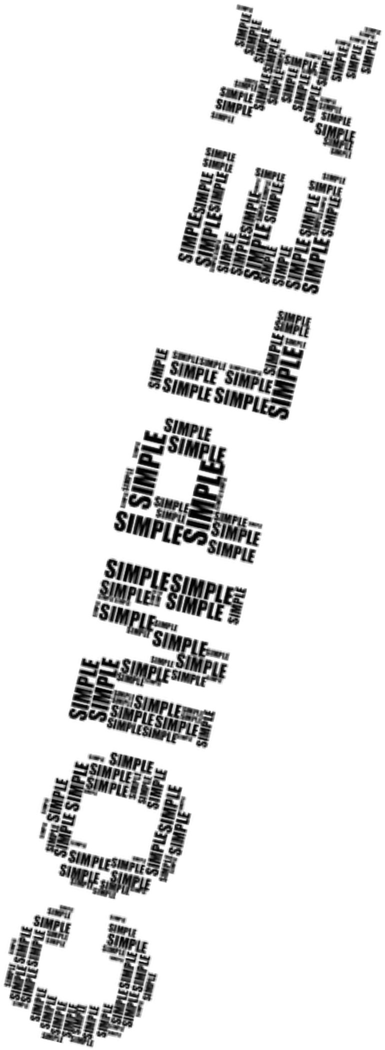



**[Kra2017a]** Kratzke, N. (2017). Smuggling Multi-Cloud Support into Cloud-native Applications using Elastic Container Platforms. In *Proceedings of the 7th Int. Conf. on Cloud Computing and Services Science (CLOSER 2017)* (pp. 29–42).

**[QK2018a]** Quint, P.-C., & Kratzke, N. (2018). Towards a Lightweight Multi-Cloud DSL for Elastic and Transferable Cloud-native Applications. In *Proceedings of the 8th Int. Conf. on Cloud Computing and Services Science (CLOSER 2018, Madeira, Portugal)*.

# Do we need really further languages?






- Cloud applications are composed of distributed processes (deployment units/containers)
    - that are operated elastically (horizontal scaling)
    - operated on different platforms and infrastructures
    - and each deployment unit maybe developed using different programming languages (polyglot).
- The units are getting simpler (microservice style).
- But complexity never disappears. It is just hidden/capsuled.
- If the units are getting simpler, the integration should tend to get more complicated.

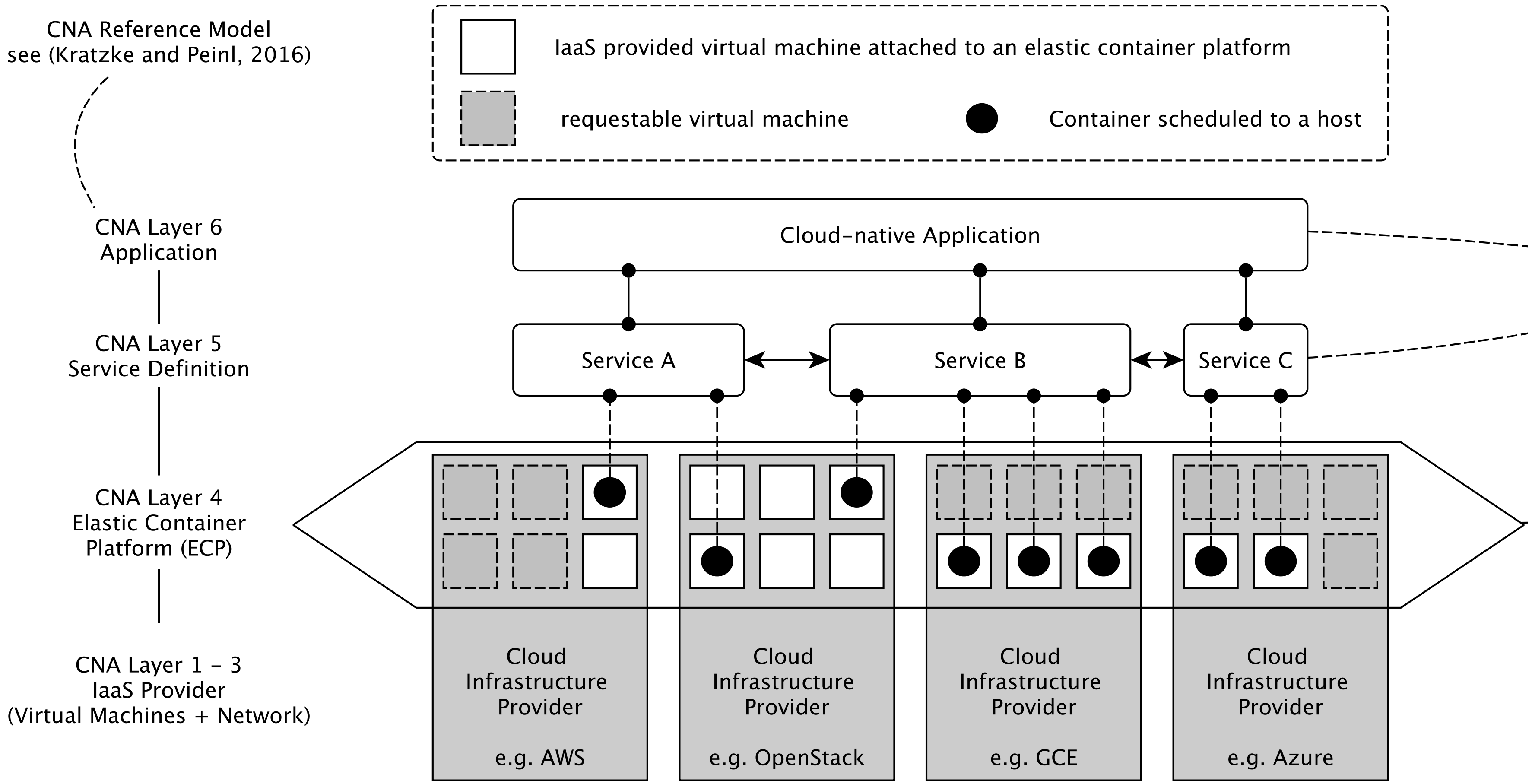
# Cloud-native Application Definition

A **cloud-native application** is a distributed, elastic and horizontal scalable system composed of (micro)services which isolates state in a minimum of stateful components. The application and each self-contained deployment unit of that application is designed according to cloud-focused design patterns and operated on a self-service elastic platform.

**[KQ2017a]** Kratzke, N., & Quint, P.-C. (2017). Understanding Cloud-native Applications after 10 Years of Cloud Computing - A Systematic Mapping Study. *Journal of Systems and Software*, *126* (April).

# We need some guidance ...
## ClouNS – Cloud-native Application Reference Model

FACH HOCHSCHULE LÜBECK




**[KP2016]** Kratzke, N., & Peinl, R. (2016). ClouNS - a Cloud-Native Application Reference Model for Enterprise Architects. In *2016 IEEE 20th International Enterprise Distributed Object Computing Workshop (EDOCW)* (pp. 1–10).

**[QK2018a]** Quint, P.-C., & Kratzke, N. (2018). Towards a Lightweight Multi-Cloud DSL for Elastic and Transferable Cloud-native Applications. In *Proceedings of the 8th Int. Conf. on Cloud Computing and Services Science (CLOSER 2018, Madeira, Portugal)*.

# So, what would a cloud programming language be?

FACH HOCHSCHULE LÜBECK
University of Applied Sciences

*A **computer** programming language is a notation used to write **computer programs**, which involves a **computer** performing some kind of **computation or algorithm** and possibly control of **external devices** such as printers, disk drives, and so on.*

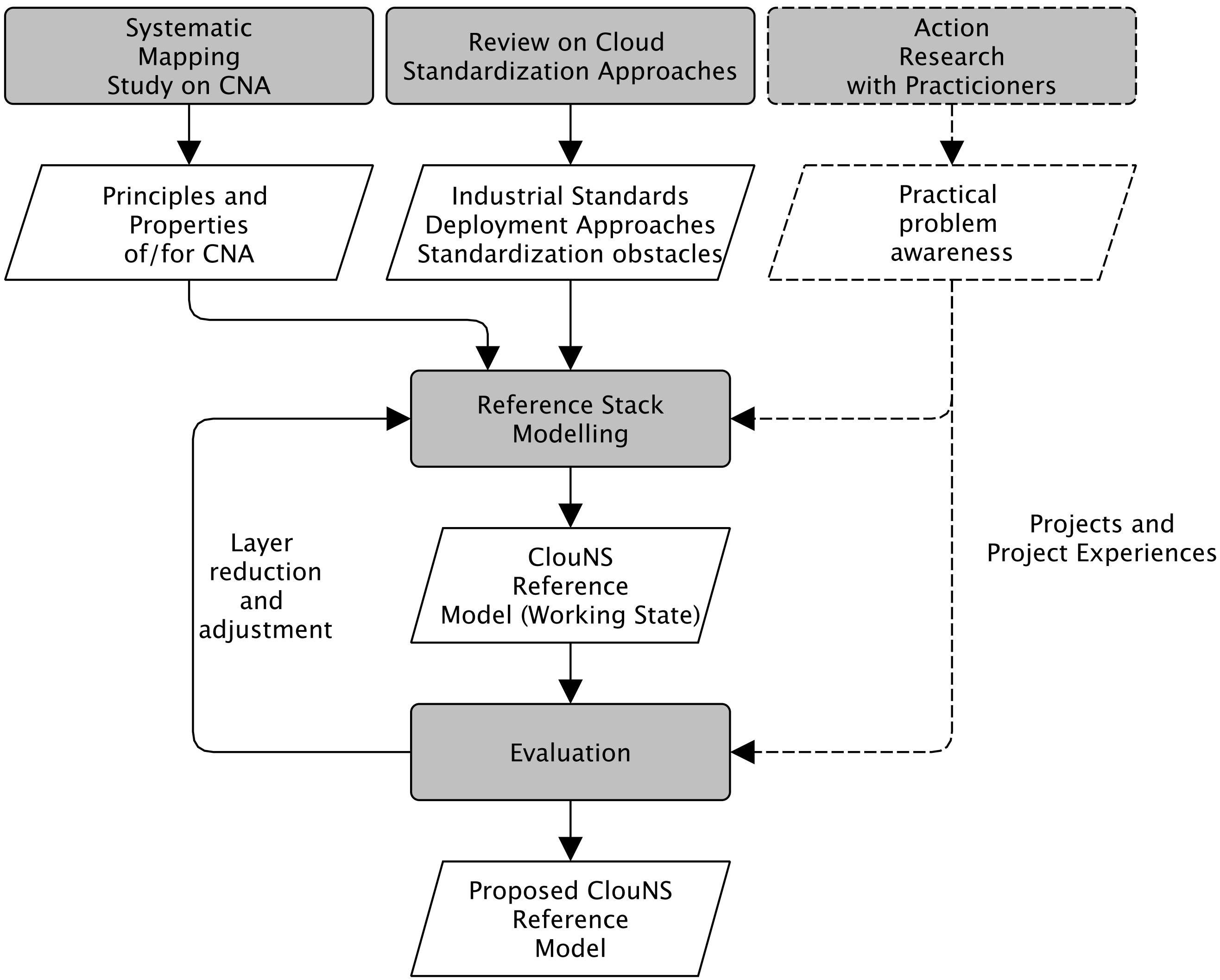
Adapted from ACM SIGPLAN/Wikipedia

*A **cloud** programming language is a notation used to define **cloud applications** to be provided via **cloud infrastructures or platforms** performing **processes** and possibly composing further **internal and external services** such as authentication, scaling, storage, messaging, logging, and further domain/problem specific services.*

# Research Methodology

## Understanding CNA after 10 Years of Cloud Computing

FACH
HOCHSCHULE
LÜBECK
University of Applied Sciences




CNA == Cloud-native Application

# Cloud-native Application



Often heard by practitioners: *„A cloud-native application is an application intentionally designed for the cloud.*" True, but helpful?

| What? Be IDEAL | Why? There is a need for .. | How? Integrate ... |
| --- | --- | --- |
| • **I**solated State<br>• **D**istributed<br>• **E**lastic<br>• **A**utomated management<br>• **L**oosely coupled | • Speed (delivery)<br>• Safety (fault tolerance, design for failure)<br>• Scalability<br>• Client diversity | • (Micro)service oriented architectures (M)SOA<br>• Use API-based collaboration<br>• Consider cloud-focused pattern catalogues<br>• Use self-service agile platforms |

*C. Fehling, F. Leymann, R. Retter, W. Schupeck, and P. Arbitter, **Cloud Computing Patterns: Fundamentals to Design, Build, and Manage Cloud Applications**. Springer, 2014.*

*M. Stine, **Migrating to Cloud-Native Application Architectures**. O'Reilly, 2015*

A. Balalaie, A. Heydarnoori, and P. Jamshidi, "**Migrating to Cloud-Native Architectures Using Microservices**", CloudWay 2015, Taormina, Italy

S. Newman, **Building Microservices**. O'Reilly, 2015.

# Cloud Application Maturity Level



TABLE III. **Cloud Application Maturity Model**, adapted from *OPEN DATA CENTER ALLIANCE Best Practices [12]*

| Level | Maturity | Criteria |
|---|---|---|
| 3 | Cloud native | - A CNA can migrate across infrastructure providers at runtime and without interruption of service.<br>- A CNA can automatically scale out/in based on stimuli. |
| 2 | Cloud resilient | - The application state is isolated in a minimum of services.<br>- The application is unaffected by dependent service failures.<br>- The application is infrastructure agnostic. |
| 1 | Cloud friendly | - The application is composed of loosely coupled services.<br>- Application services are discoverable by name (not by IP).<br>- Application components are designed using cloud patterns.<br>- Application compute and storage are separated. |
| 0 | Cloud ready | - The application runs on virtualized infrastructure.<br>- The application can be instantiated from image or script. |

# Panel on CLOUD COMPUTING

## Challenges in Cloud Computing-based Systems

## "Commonalities & Barriers"

CloudComp 2018

Raimund K. Ege

Computer Science, Northern Illinois University, USA

ege@niu.edu

2/21/2018

# Internet vs. Cloud





- Connectivity of hosts
- Hosts provide capabilities in groups

- Slicing vs. portioning

- Host/connectivity relies on:
  - Technology
  - Standards

# Cloud Platforms

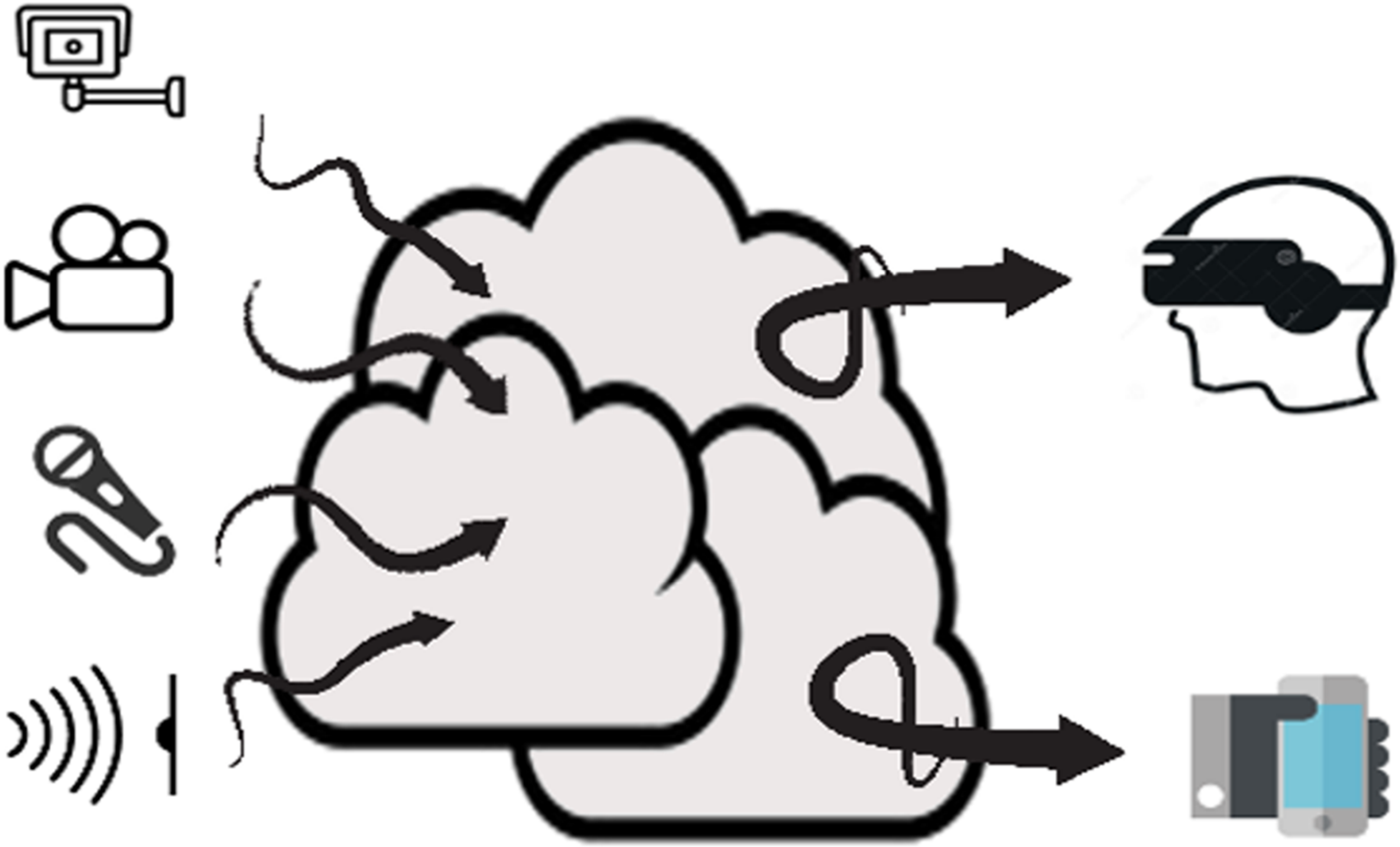


- Amazon Web Services
- Microsoft Azure
- Google Cloud Platform
- ... many others

- Also:
  - Heroku

# Big Picture Idea



# Commonalities & Barriers





- Hosting
  - Static file server
  - World-wide: region-based, duplicated
- Web application server
  - ASP.Net C#, Java Servlet Container
  - PHP, Ruby, Python, …
- Middleware
  - Security, authentication
  - RESTful services
- Database

# Cloud computing 2018
# Panel on CLOUD/SERVICES

## Challenges in Cloud Computing-based Systems

## Real Time Cloud Computing

**Panelist**

**Yong Woo LEE, Ph.D.**

**Professor, University of Seoul**

**President, Smart Consortium for Seoul, Korea**

**Chair, Seoul Grid Center**

**Chair, The Korean National Committee for ISO JTC1/SC22**

**2018. 2. 21, Barcelona**

# Real-time cloud computing

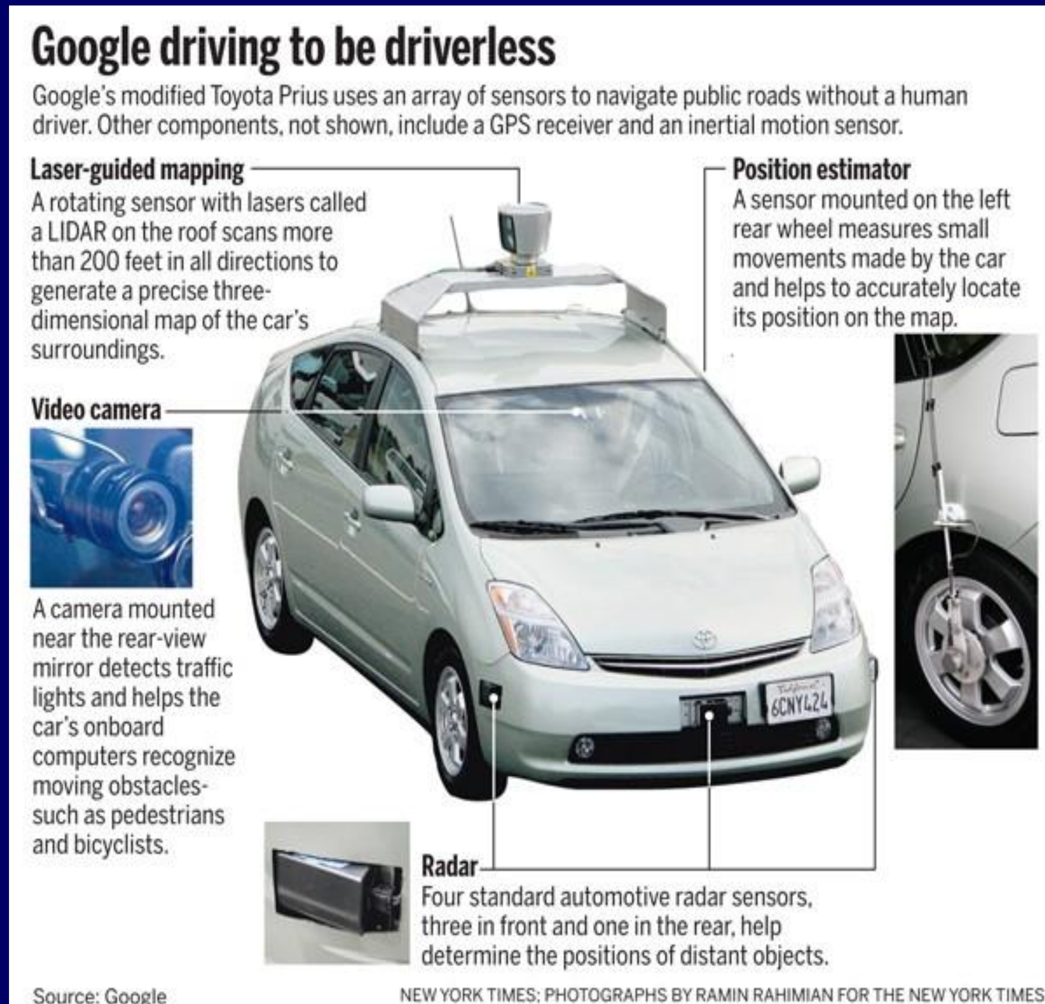
**Presented for Cloud Computing 2018**

February 21, 2018
Barcelona

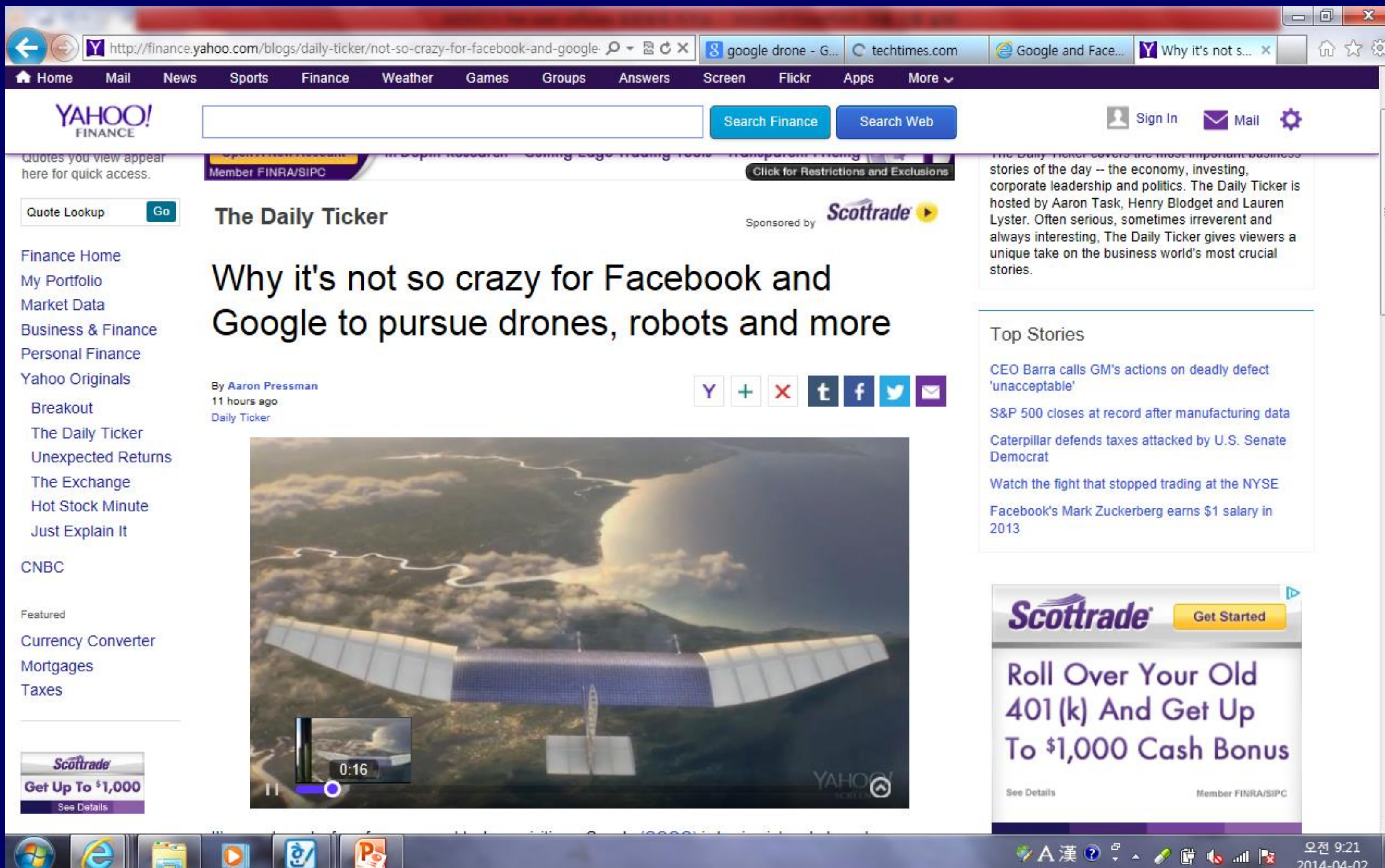
# Real Time Cloud Computing

- Essential for the Mobile computing in many cases.
- – Virtual reality
- – Smart cars.
- – Drones
- – IoT
- – Robots
- – etc.

# Smart cars


Google driving to be driverless
Google's modified Toyota Prius uses an array of sensors to navigate public roads without a human driver. Other components, not shown, include a GPS receiver and an inertial motion sensor.
Laser-guided mapping
A rotating sensor with lasers called a LIDAR on the roof scans more than 200 feet in all directions to generate a precise three-dimensional map of the car's surroundings.
Position estimator
A sensor mounted on the left rear wheel measures small movements made by the car and helps to accurately locate its position on the map.
Video camera
A camera mounted near the rear-view mirror detects traffic lights and helps the car's onboard computers recognize moving obstacles- such as pedestrians and bicyclists.
6CNY424
Radar
Four standard automotive radar sensors, three in front and one in the rear, help determine the positions of distant objects.
Source: Google
NEW YORK TIMES; PHOTOGRAPHS BY RAMIN RAHIMIAN FOR THE NEW YORK TIMES

# Drones


http://finance.yahoo.com/blogs/daily-ticker/not-so-crazy-for-facebook-and-google
google drone - G...
techtimes.com
Google and Face...
Why it's not s...
Home
Mail
News
Sports
Finance
Weather
Games
Groups
Answers
Screen
Flickr
Apps
More
YAHOO! FINANCE
Search Finance
Search Web
Sign In
Mail
here for quick access.
Quote Lookup
Go
Finance Home
My Portfolio
Market Data
Business & Finance
Personal Finance
Yahoo Originals
Breakout
The Daily Ticker
Unexpected Returns
The Exchange
Hot Stock Minute
Just Explain It
CNBC
Featured
Currency Converter
Mortgages
Taxes
Scottrade
Get Up To $1,000
See Details
Member FINRA/SIPC
Click for Restrictions and Exclusions
The Daily Ticker
Sponsored by Scottrade
Why it's not so crazy for Facebook and Google to pursue drones, robots and more
By Aaron Pressman
11 hours ago
Daily Ticker
0:16
stories of the day -- the economy, investing, corporate leadership and politics. The Daily Ticker is hosted by Aaron Task, Henry Blodget and Lauren Lyster. Often serious, sometimes irreverent and always interesting, The Daily Ticker gives viewers a unique take on the business world's most crucial stories.
Top Stories
CEO Barra calls GM's actions on deadly defect 'unacceptable'
S&P 500 closes at record after manufacturing data
Caterpillar defends taxes attacked by U.S. Senate Democrat
Watch the fight that stopped trading at the NYSE
Facebook's Mark Zuckerberg earns $1 salary in 2013
Scottrade
Get Started
Roll Over Your Old 401(k) And Get Up To $1,000 Cash Bonus
See Details
Member FINRA/SIPC
오전 9:21
2014-04-02

# Smart Devices.





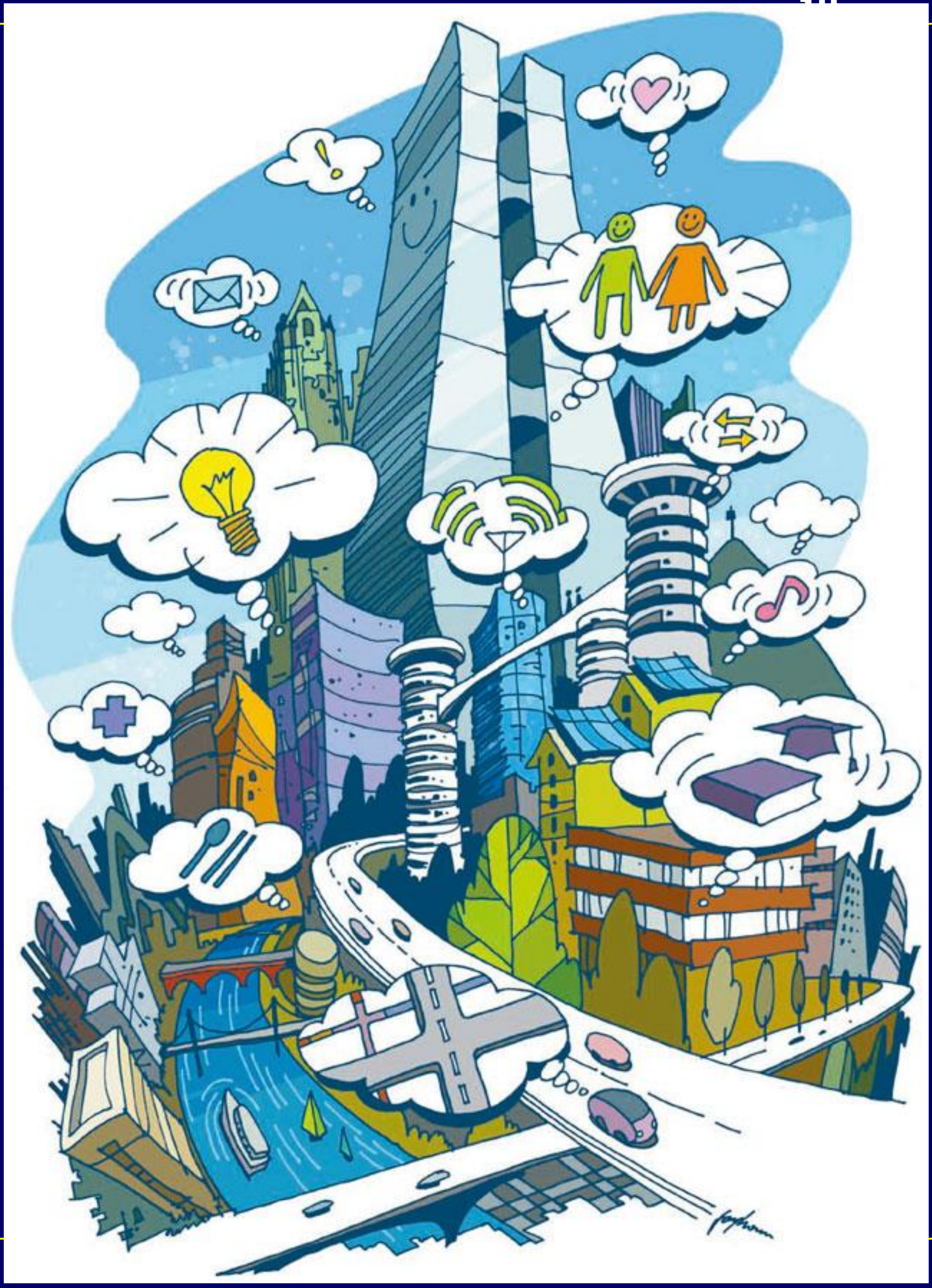

# IoT by iStockphoto/chris_lemmens

# The Smart society with smart robots.

# Real Time Cloud Computing

- Essential for real time processing in smart cities.
- Essential for streamed data processing.
- Essential for stream reasoning.

# Smart-Cities



# 5 G mobile communication enables real-time cloud computing in mobile devices.

# 5 G Mobile computing

## WMC Barcelona 2016.2

# 5 G Mobile computing

## WMC Barcelona 2016.2

# 5 G Mobile computing

## WMC Barcelona 2016.2

# 5 G Mobile computing

## WMC Barcelona 2018.2

# Conclusion


תודה
Dankie
Gracias
Спасибо
شكراً
Merci
Takk
Köszönjük
Terima kasih
Grazie
Dziękujemy
Dėkojame
Ďakujeme
Vielen Dank
Paldies
Kiitos
Täname teid
谢谢
Thank You
Tak
感謝您
Obrigado
Teşekkür Ederiz
감사합니다
Σας Ευχαριστούμ
ขอบคุณ
Bedankt
Děkujeme vám
ありがとうございます
Tack