



Platform-as-a-Service és Telekom felhő

Csaba Simon

Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Távközlési és Médiainformatikai Tanszék

BME-TMIT

INTRO – CLOUDS REVISITED

Cloud business roles

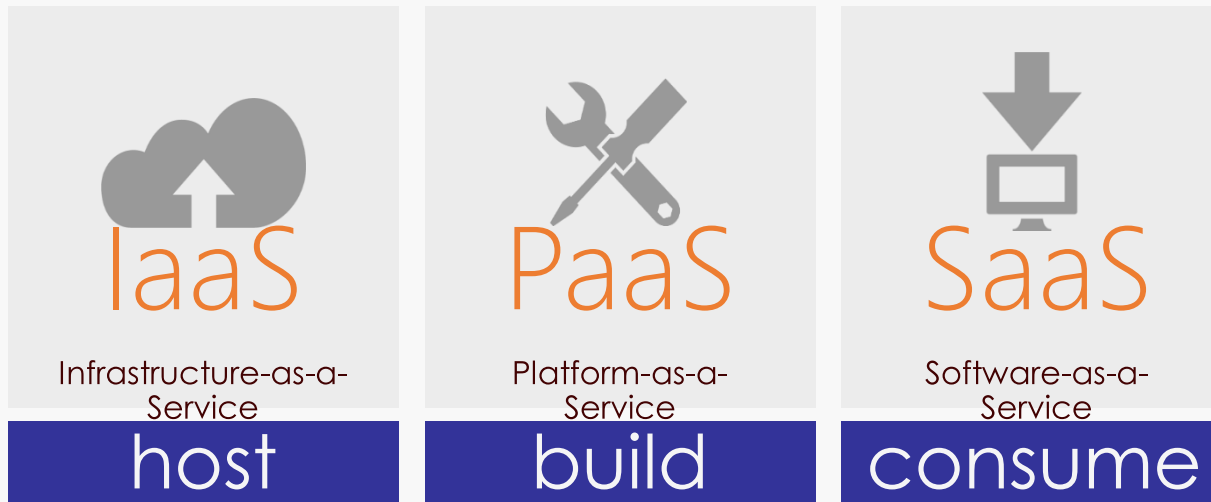
Expected benefits



Required infrastructure features



Cloud terminology – technological view



Cloud types

Deployment mode

Private Cloud

Public Cloud

Hybrid Cloud

Service mode

Infrastructure
as a Service
(IaaS)

Platform as a
Service
PaaS

Software as a
Service
SaaS

WHAT IS THE PAAS?

Platform as a Service – the **services**

Services: what offers PaaS to us?

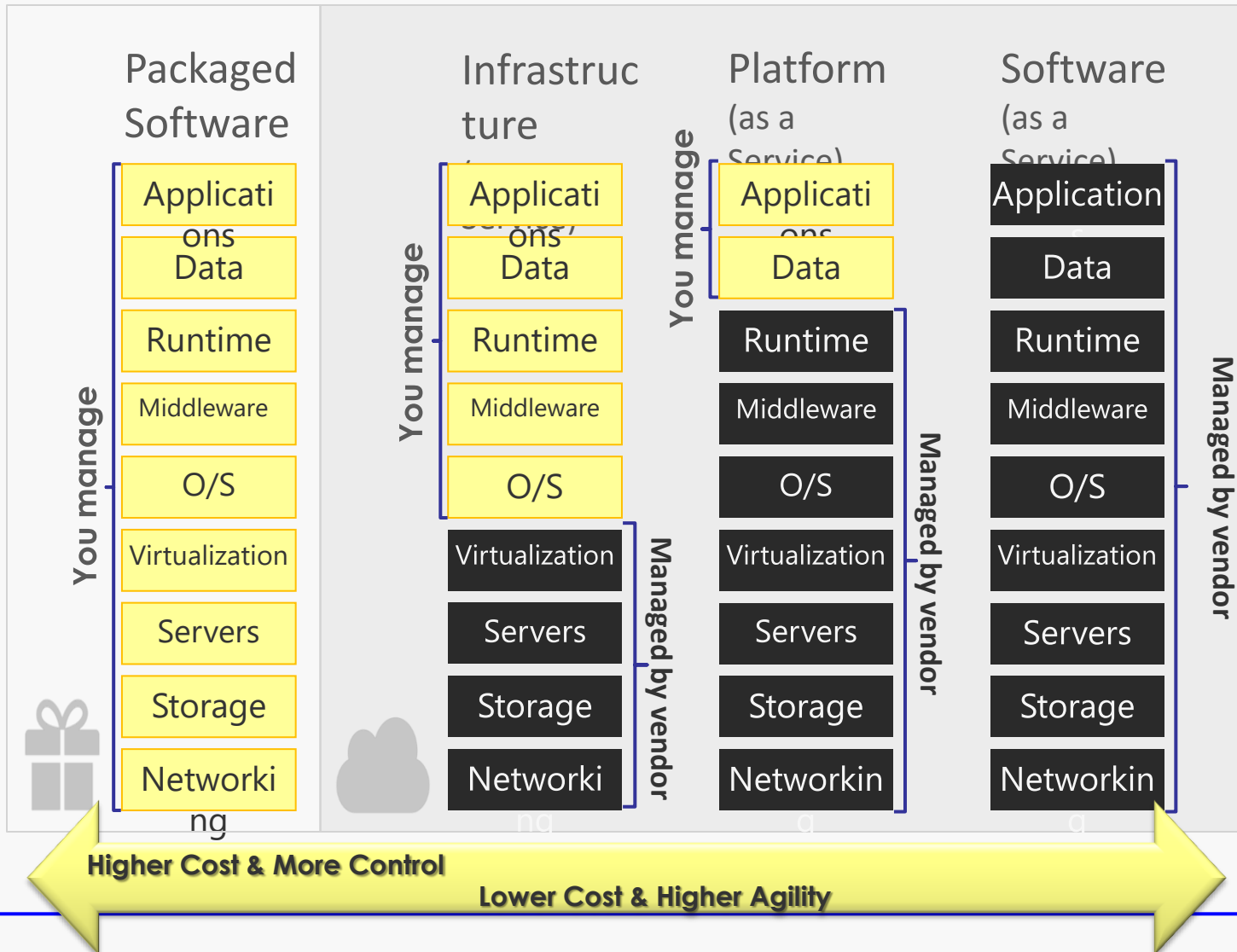
- App **deployment**
- **Scaling** (horizontal, vertical, auto)
- Load **balancing**
- Health **monitoring**, auto recovery
- **Logging** service
- External/internal services, **marketplace**

Advantages of PaaS – the **Platform** assures

- deployment
- load balancing
- high availability
- log aggregation
- scaling
- Image mgmt –
 - » Libraries, kernel versions
 - » Security updates

(Tenant's problem: just develop your **own app**)
- **Előny**: A felhasználó a saját alkalmazásának kidolgozására koncentrálnak

Compare the *aaS-es



Comparing the *aaS models

- » IaaS: The end user maintains control of the operating system and applications on the hardware.
- » PaaS: end user has to development, testing, deployment, and ongoing maintenance of applications
- » SaaS: end users pay on a per-use basis

PaaS in a nutshell

Allround and complex set of services

Fast application management

- build
- telepítés
- menedzsment

On global (multi-site) networks



Flexible



Open



Solid

PaaS system varieties

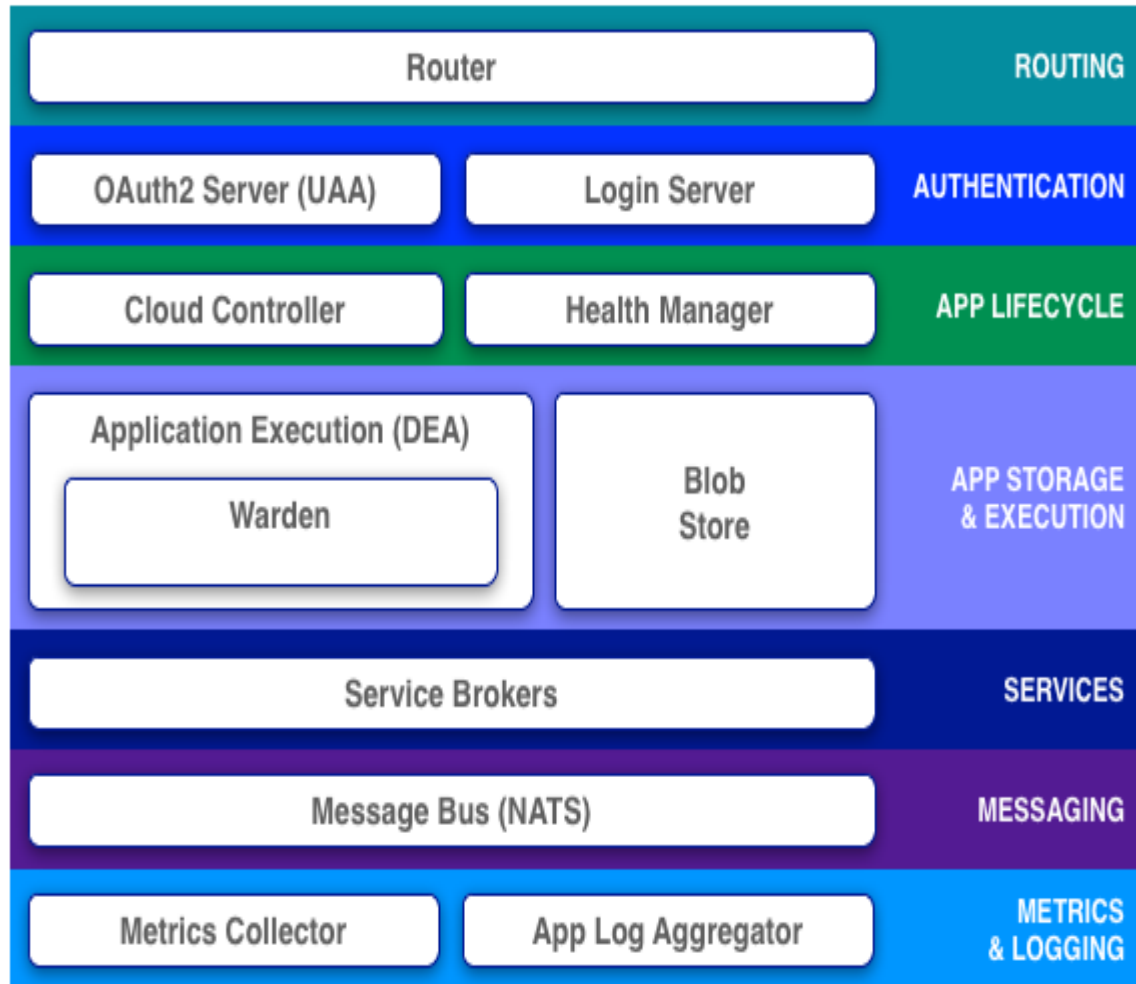
- Services that help develop and test apps
 - infrastructure is maintained by the provider
- Reduced infrastructure complexity
 - more effective overall application development
- Runtime environments are usually lock-in free
 - but might create lock-ins to provider specific infrastructure
- Usually simple network topology and access control
 - build your services as they would be open to the Internet
- The features and services provided vary a lot
 - from simple customizable runtime (CloudFoundry) to full marketplace of services (Heroku)

PaaS - Toolchain as a Service

- Manage your project
 - Trello, Jira OnDemand, Sprint.ly, PivotalTracker, ...
- Create your code
 - Cloud9, Koding, Nitrous, ...
- Host your code
 - GitHub, Bitbucket, ...
- Build your code
 - Codeship, Travis CI, CloudBees, Drone, ...
- Test your code
 - BrowserStack, Sauce Labs, Xamarin Test Cloud, Blitz, ...
- Distribute your code
 - npm, Bintray, Maven Central, PyPI, Docker Hub, ...

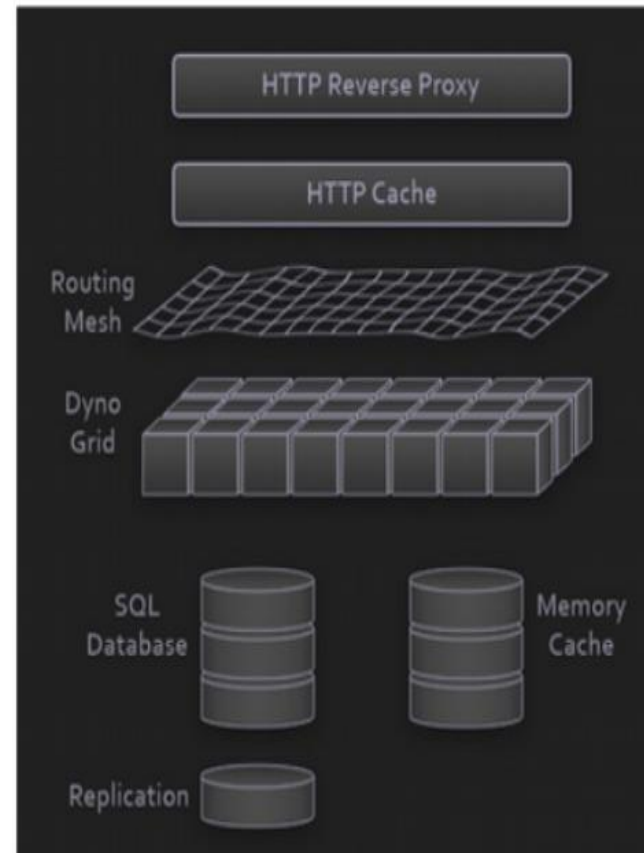
PAAS ARCHITECTURE

Cloud Foundry - Architecture



Heroku architecture

- Reverse Proxy by nginx
 - terminates SSL
 - forwards to cache layer
- HTTP Cache by Varnish
 - returns cached pages immediately
 - forwards to routing mesh
- Routing Mesh written in Erlang
 - routes to an existing dyno
 - spawns a dyno if none available
- Dyno Grid ('railgun' servers)
 - AWS hosted EC2 instances
 - multiple dynos per server



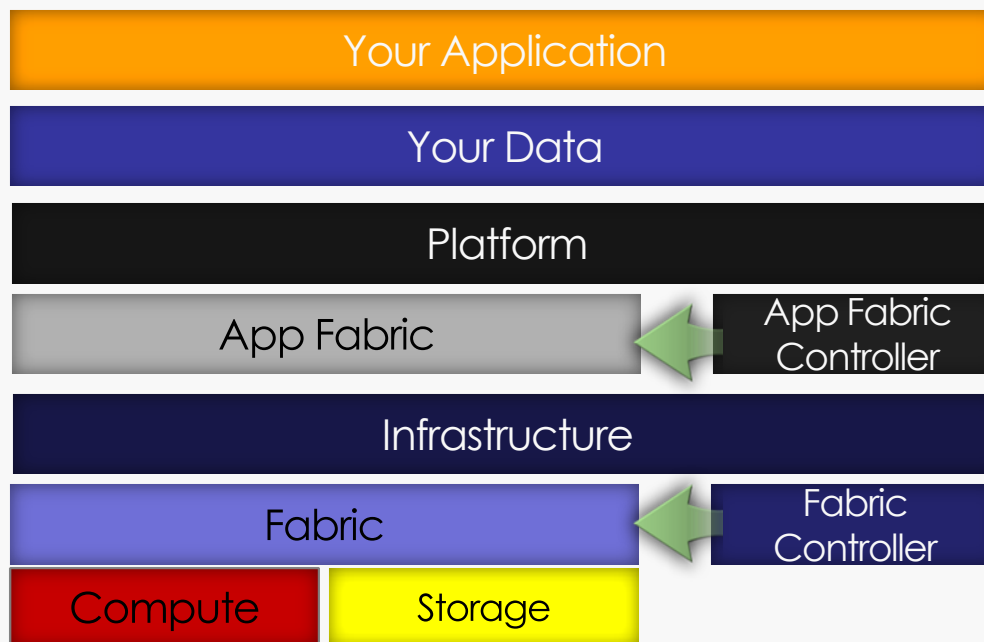
by David Feng / CC BY-NC-SA 2.0

Heroku terminology

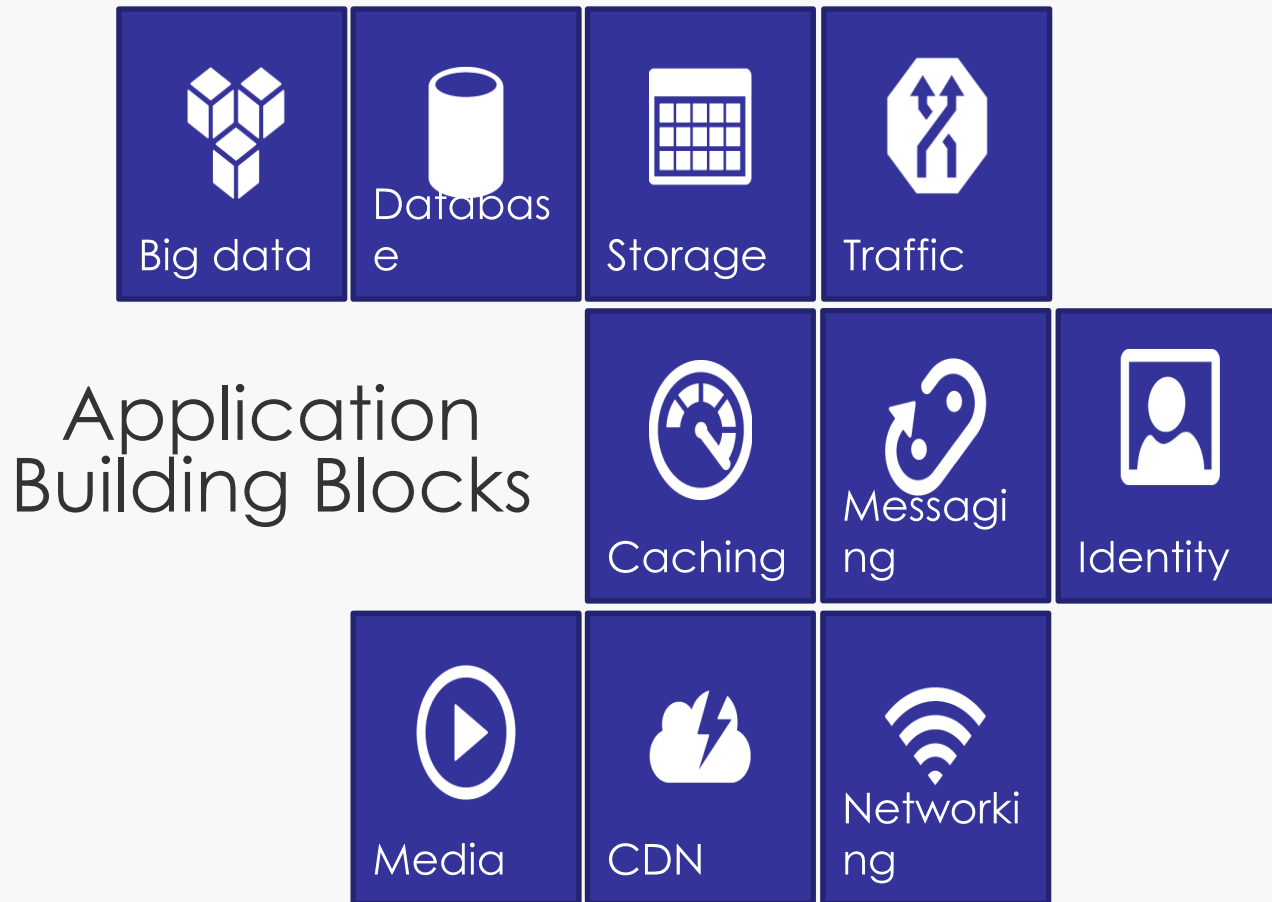
- **Application** source code and description of any dependencies
- **Procfile** list of process types – named commands to be executed
- **Deployment** sending application to Heroku using git or dropbox
- **Buildpack** compilation process that creates a slug from application
- **Slug** bundle of application, language runtime and compilation output
- **Dyno** isolated, virtualized Linux container for application runtime
- **Release** append-only ledger of slugs, config vars and add-ons
- **Config var** configuration data hosted independently of source code
- **Add-on** easily attachable third party cloud services
- **Logplex** collates logs from all running dynos and other components

Winazure

Windows Azure™



Azure Building Block Services



Sharp competition, „friendly” competitors

- » Stealing customers from a PaaS
 - » Migrate them to a different (IaaS) provider
 - » „By chance” it also offers to be the new PaaS supplier in this new setup
- » <https://www.skyliner.io/offer/heroku>

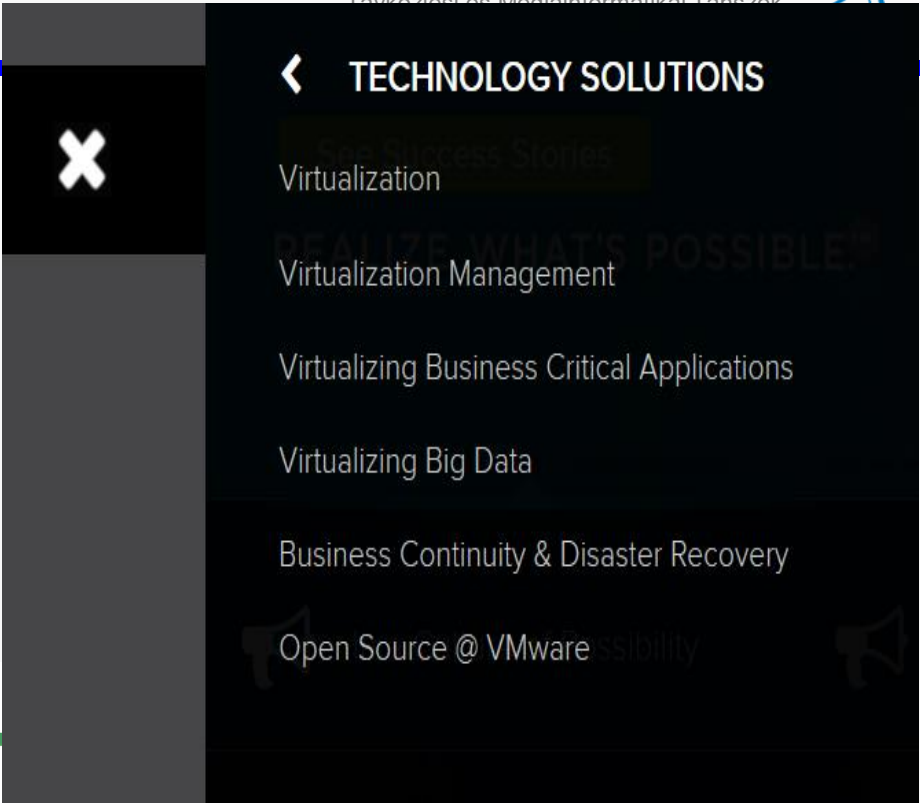
Skyliner



We'll port your Heroku app to AWS for free

Skyliner is a Heroku-like launch platform that you can use on AWS. Switching to Skyliner will save you a bundle on your infrastructure.

Pioneer of enterprise level virtualization



vmware


☰ Pivotal-VMware Cloud-Native Stack




Pivotal Cloud Foundry – VMware Photon Platform Deliver

 Best-in-Class Cloud-Native Stack

VMware is paired with Pivotal Cloud Foundry to deliver startup speed with enterprise reliability.

 Simple to Purchase, Install and Maintain

 Faster Innovation for Developers and Operations

The joint solution is built for speed, scale and programmability, making it usable by developers, operations, and everyone in-between. It is accessible and controllable via APIs, CLI for developers, and a GUI for operations teams.

TELCO GRADE PAAS

PaaS – what is missing,

Telecom requirements: PaaS is coming from the IT (web) world

- Usually there is a loadbalancer for HTTP only
 - Other protocols (SIP, diameter, TCP session) are not supported
- Internal state info to be stored in external DB/cache
 - Performance issues
- No guarantee on QoS/ response times
- VMs and networks completely hidden
 - Colocation of VMs that communicate a lot with each other?
 - No networkin optimization (e.g., intel DPDK)
- No standard PaaS
 - Migration from one operator to the other?

Requirements for a PaaS

- » What does it take to write a „PaaS-compatible” app?
- » Actually it is the same for any SaaS, PaaS,... app
- » <http://12factor.net/>

PaaS criteria 1/2

- » **I. Codebase**
- » **One codebase tracked in revision control, many deploys**
- » **II. Dependencies**
- » **Explicitly declare and isolate dependencies**
- » **III. Config**
- » **Store config in the environment**
- » **IV. Backing Services**
- » **Treat backing services as attached resources**
- » **V. Build, release, run**
- » **Strictly separate build and run stages**
- » **VI. Processes**
- » **Execute the app as one or more stateless processes**

PaaS criteria 2/2

- » **VII. Port binding**
- » **Export services via port binding**
- » **VIII. Concurrency**
- » **Scale out via the process model**
- » **IX. Disposability**
- » **Maximize robustness with fast startup and graceful shutdown**
- » **X. Dev/prod parity**
- » **Keep development, staging, and production as similar as possible**
- » **XI. Logs**
- » **Treat logs as event streams**
- » **XII. Admin processes**
- » **Run admin/management tasks as one-off processes**

Microservices

- » Micro services = software module implementing a given function
 - » Own states
 - » Multiple microservices cooperating to complete a complex task
 - » The change offered by microservice based architectures:
 - » Instead of a monolith, multiple smaller (micro) modules
 - » Scaling only the modules that run into bottlenecks
 - » Useful paradigm in telco world – deployment of a given microservices depending on the operator's needs

- » <http://martinfowler.com/articles/microservices.html>

TelcoGrade PaaS – examples

- » FeedHenry – Mbaas
- » <http://www.feedhenry.com/mobile-application-platform/mbaas/>
 - » (Twillio)
 - » <https://www.twilio.com/customers>

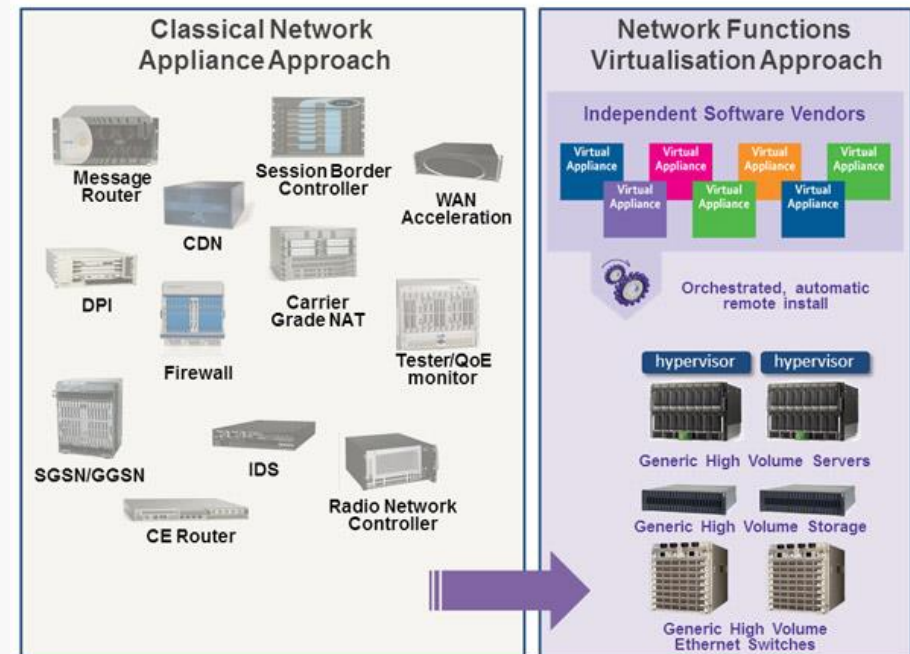
PaaS a in practice

- » PaaS based on a set of requirements
- » <http://www.paasify.it/vendors>

Network Function Virtualization

Network Functions Virtualization – NFV

- » Network Functions Virtualization – NFV
 - » Network function (e.g., cache, firewall) independent from the hardware
 - » Functionality implemented in software
 - » Running over a generic server architecture (e.g., no need for ASICs)
- » Operator point of view / Motivation
 - » Reduce the CapEx/OpEx
 - » Faster service instantiation
 - » Flexibility, adaptation
- » Standard groups
 - » ETSI NFV
 - » Open Platform for NFV (OPNFV)

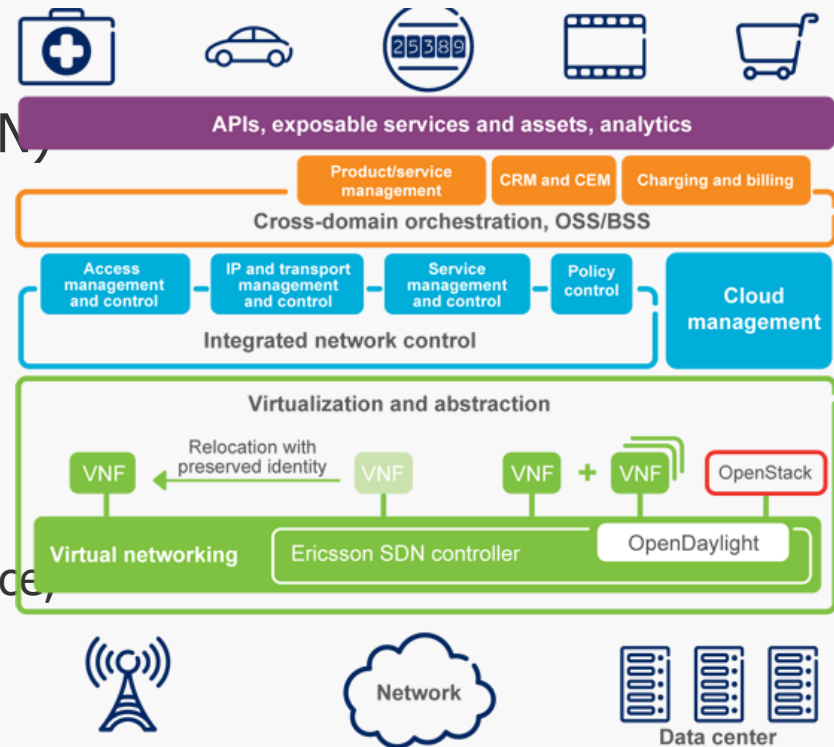


Telco cloud

- » Virtualized telco functions
 - » Packet switched core (EPC)
 - » IMS/VoLTE components (CSCF, HSS,...)
 - » Content Delivery Networks (CDN)
 - » Deep Pckt Inspection (DPI)

- » Performance
 - » Load balancing, scalability
 - » VNFs moved to end users
 - » TelCo grade service
 - » Deployment, monitoring, resilience, billing...
 - » Hardware acceleration?
 - » Virtual switch, network card

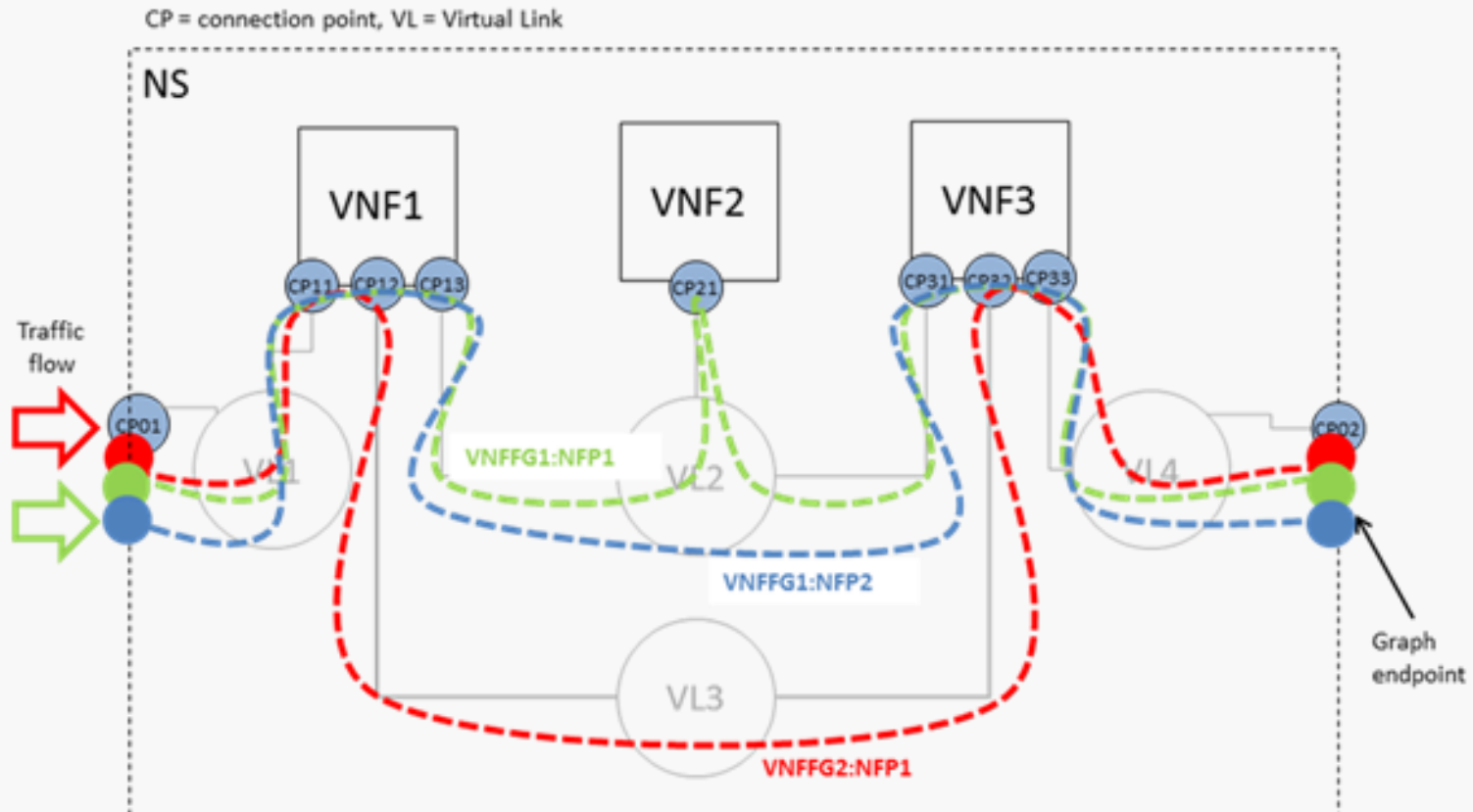
- » Ericsson: real-time cloud
 - » SDN, NFV és felhő kombinációja



Consulting, system integration and network rollout

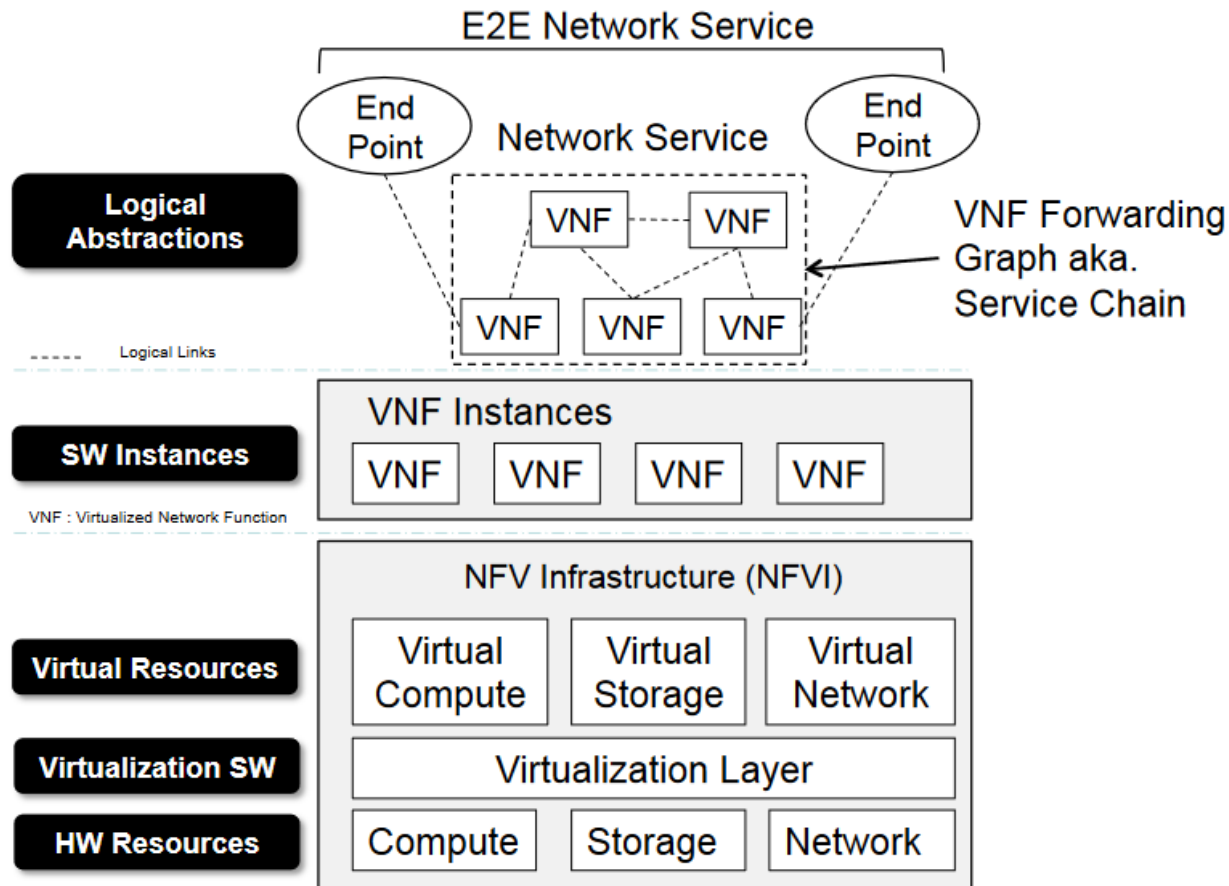
Dynamic service chaining

- » New service (NS) = linking the VNFs
 - » VNF Forwarding Graph



Dynamic service chaining

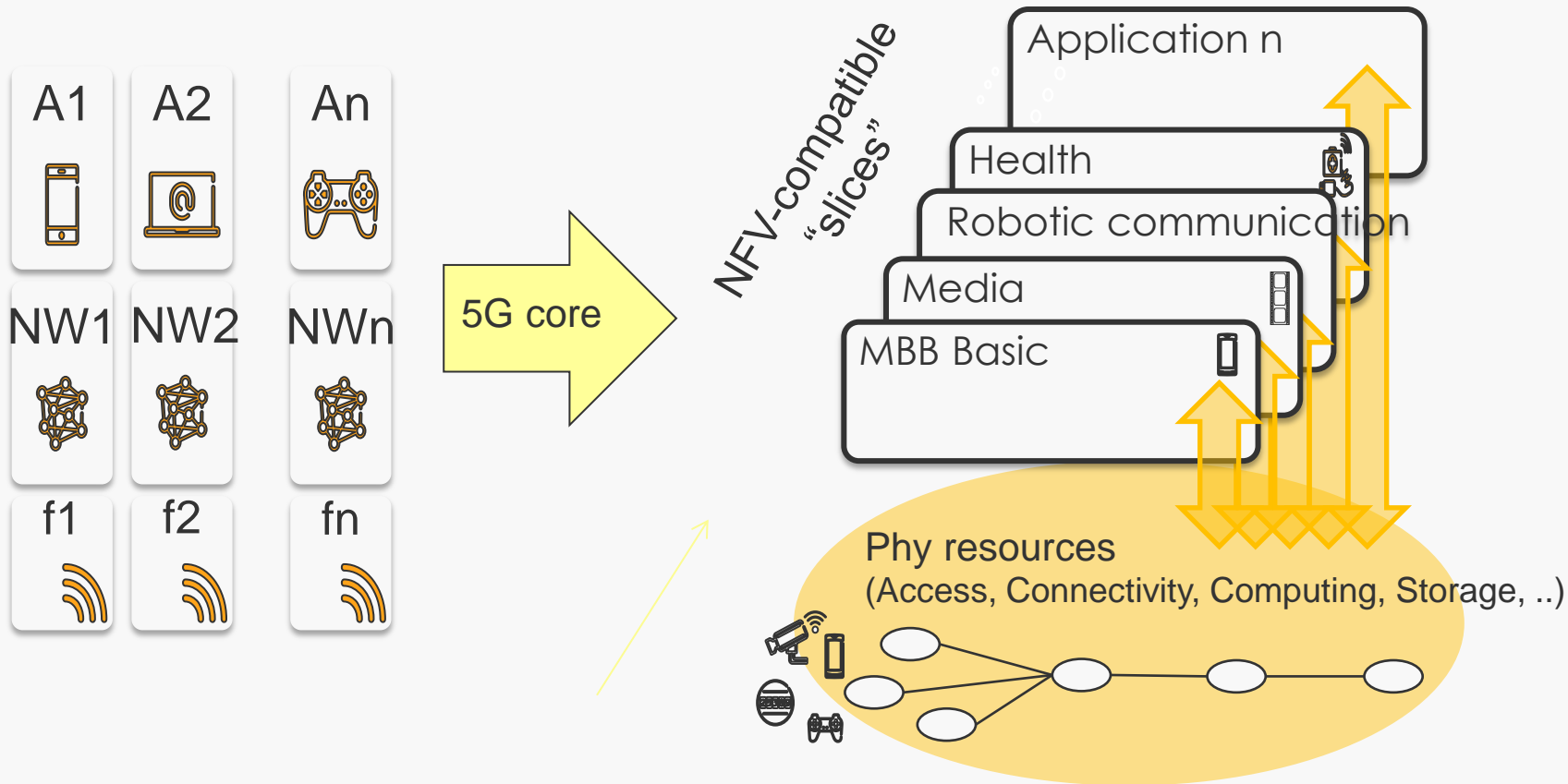
» New service = Linking the VNFs



More networks over the same infra

Dedicated physical network with own control plane and per-app resources and services

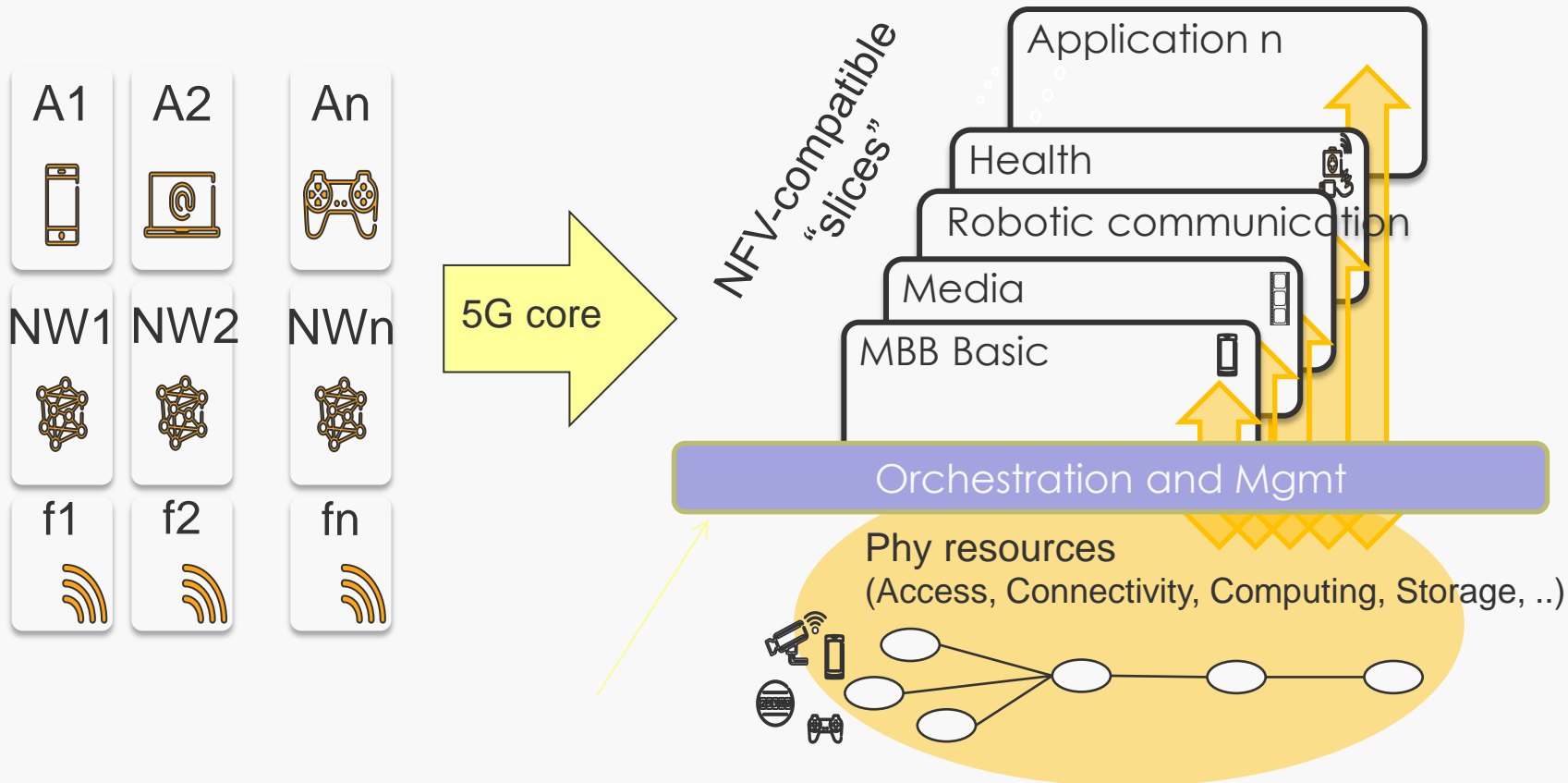
“Network factory”, market of resources and Service, abstraction levels for resources and services



More networks over the same infra

Dedicated physical network with own control plane and per-app resources and services

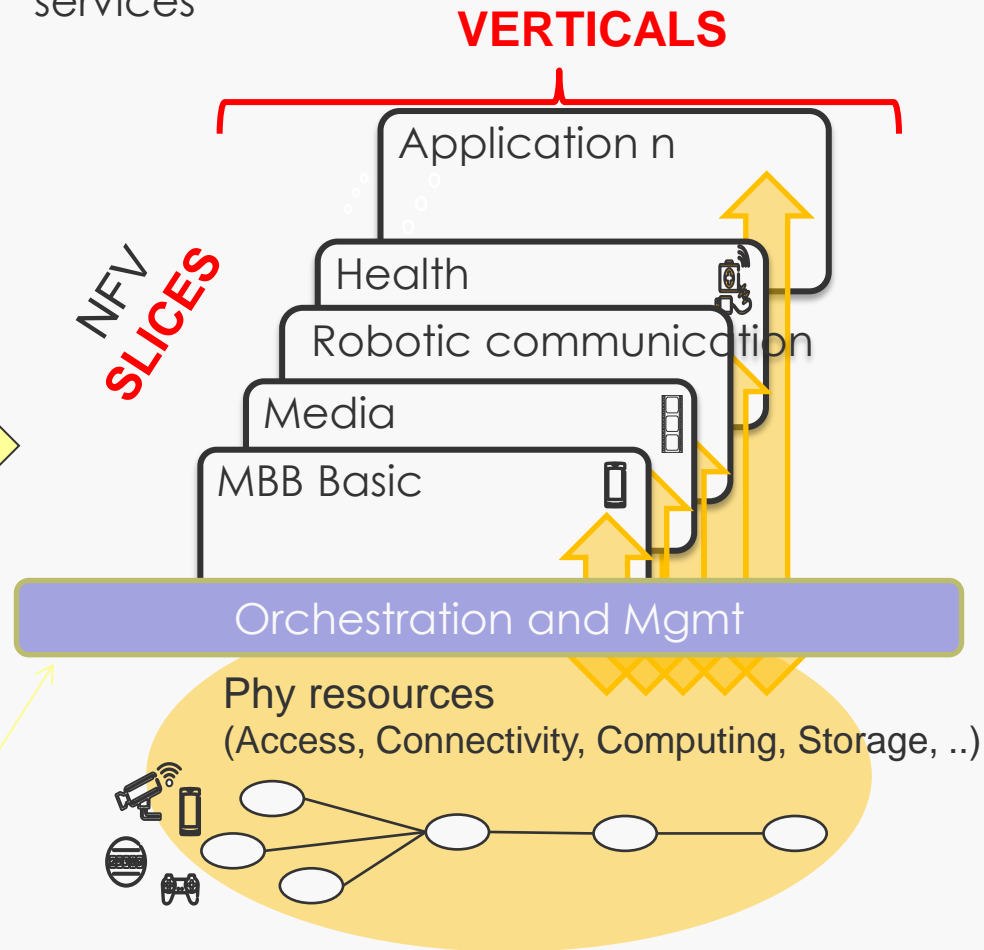
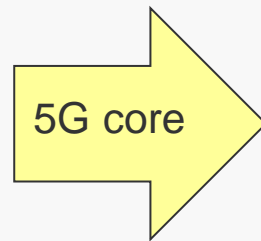
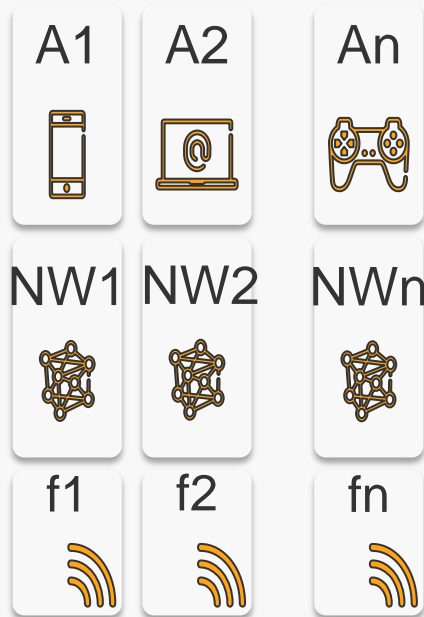
“Network factory”, market of resources and Service, abstraction levels for resources and services



More networks over the same infra

Dedicated physical network with own control plane and per-app resources and services

“Network factory”, market of resources and Service, abstraction levels for resources and services



NFV architecture

