CC2I - Introduction to PaaS

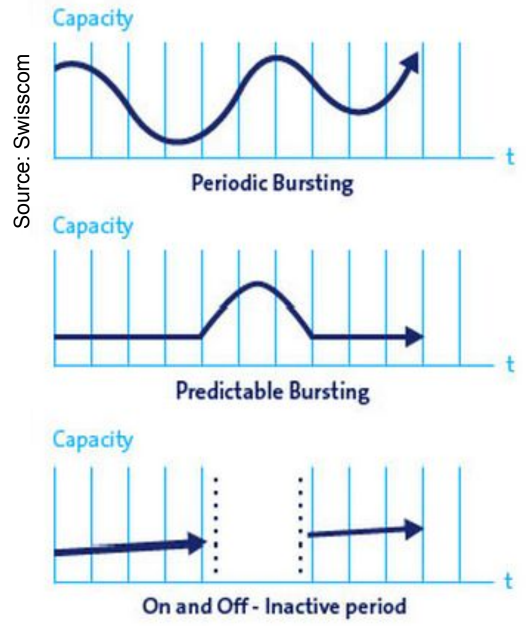
CCP2-EN Cloud Computing 2

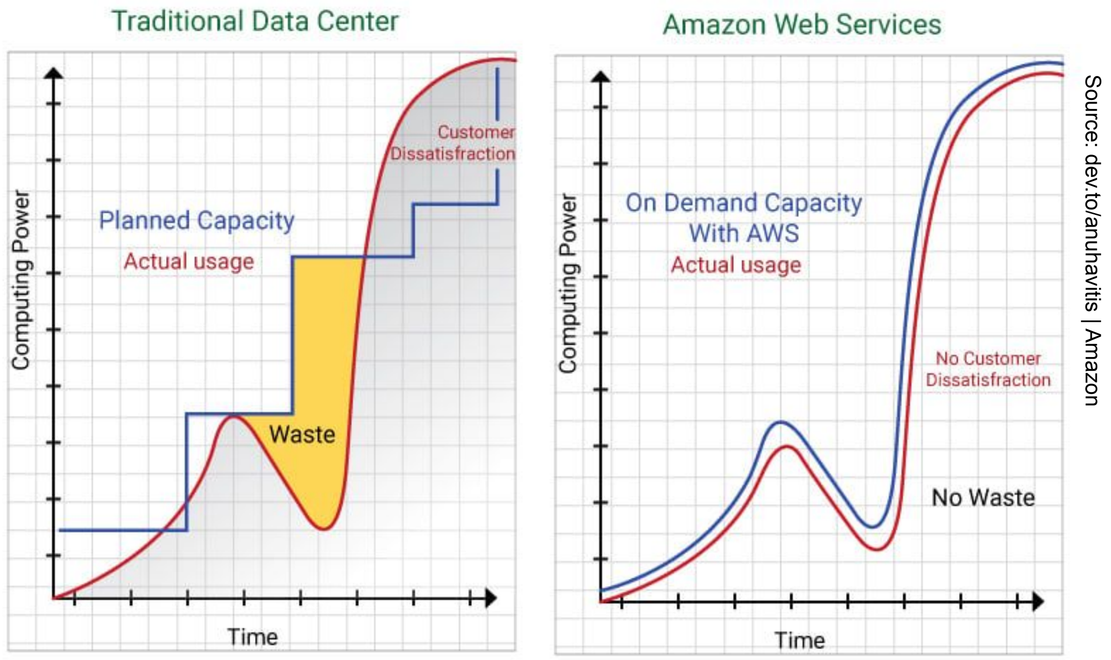
Content

- Cloud Computing Concepts (recap)
- Introduction Platform as a Service (PaaS)
- Cloud Application Platform Classification
- PaaS History and Players

Optimize ICT consumption (CCP1)

ICT consumption models





Can I consume ICT when I need it?

How much capacity do we need to provide?
Does it grow / shrink with my demands?

Zurich University of Applied Sciences
School of Engineering
InIT Institute of Applied Information Technology

Reduce Total Cost of Ownership (CCP1)

- Total Cost of Ownership (TCO)
  = Capital Expenses (CAPEX) + Operational Expenses (OPEX)

ICT related CAPEX

- Computer hardware and programs
- Network hardware and software
- Server hardware and software
- Workstation hardware and software
- Installation and integration of hardware and software
- Purchasing research
- Maintenance, warranties and support licences

ICT related OPEX

- Rented space (hosting, data center)
- Electricity (for related equipment, cooling, backup power)
- Testing costs, downtime, outage and failure expenses
- Backup and recovery process
- Technology training
- Audit (internal and external)
- Information technology personnel

What kind of ICT is needed?
Can I pay for the ICT I actually use?

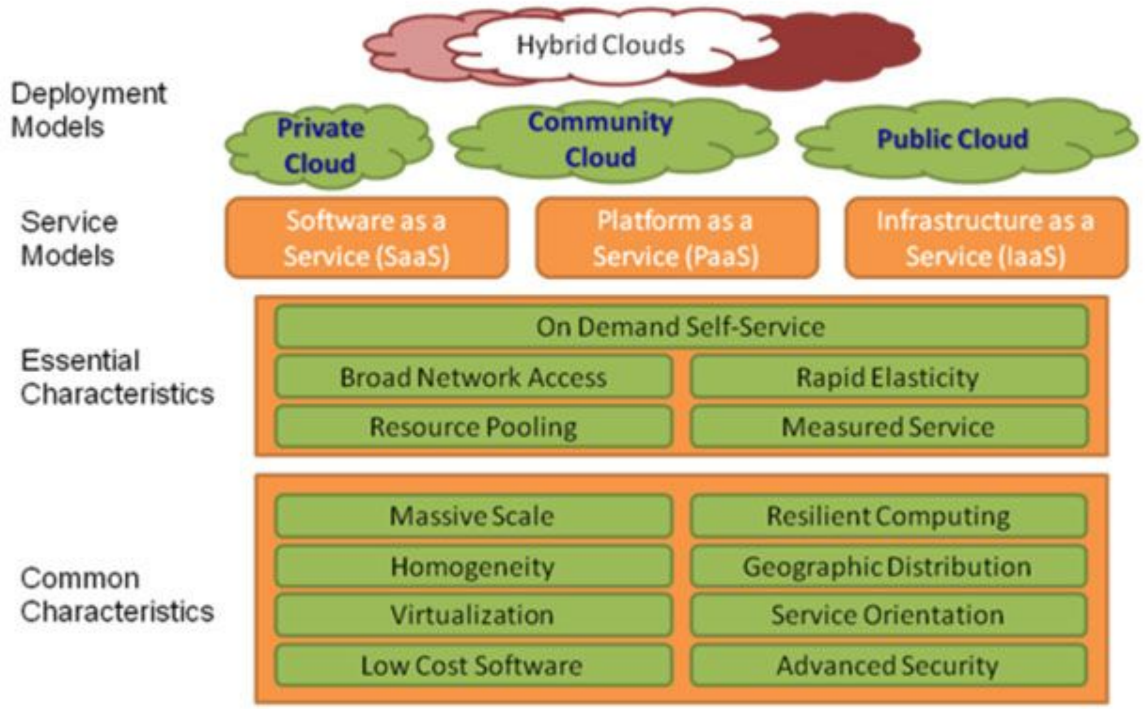
Zurich University
of Applied Sciences

School of Engineering
InIT Institute of Applied Information Technology

Conclusions (CCP1)

• How much ICT do I need? – Knowledge, skill, ability, capability, capacity, ... – Hardware, infrastructure, platform, application, …	Use service from ICT provider/expert
• Can I consume ICT when I needed it? – Now, tomorrow, in one year, ...	On-demand, self-service
• Does ICT grow with my demands? – From start-up, to SME, to enterprise, ...	Elasticity
• Does ICT shrink with my demands? – Seasons, financial crisis, solution retirement, ...	Elasticity
• Can I pay for the ICT I actually use? – Per day, per hour, per KB, per Mbit/sec, ...	Metered service, pay-as-you-go

NIST Cloud Definition Framework



Source: The NIST Cloud Computing Definition

Shift of Perspective ...

In Cloud Computing 1 (CCP1) the focus was on:

Cloud Computing from the IT Architect’s perspective

or

Design and Operation of Cloud Systems

Shift of Perspective ...

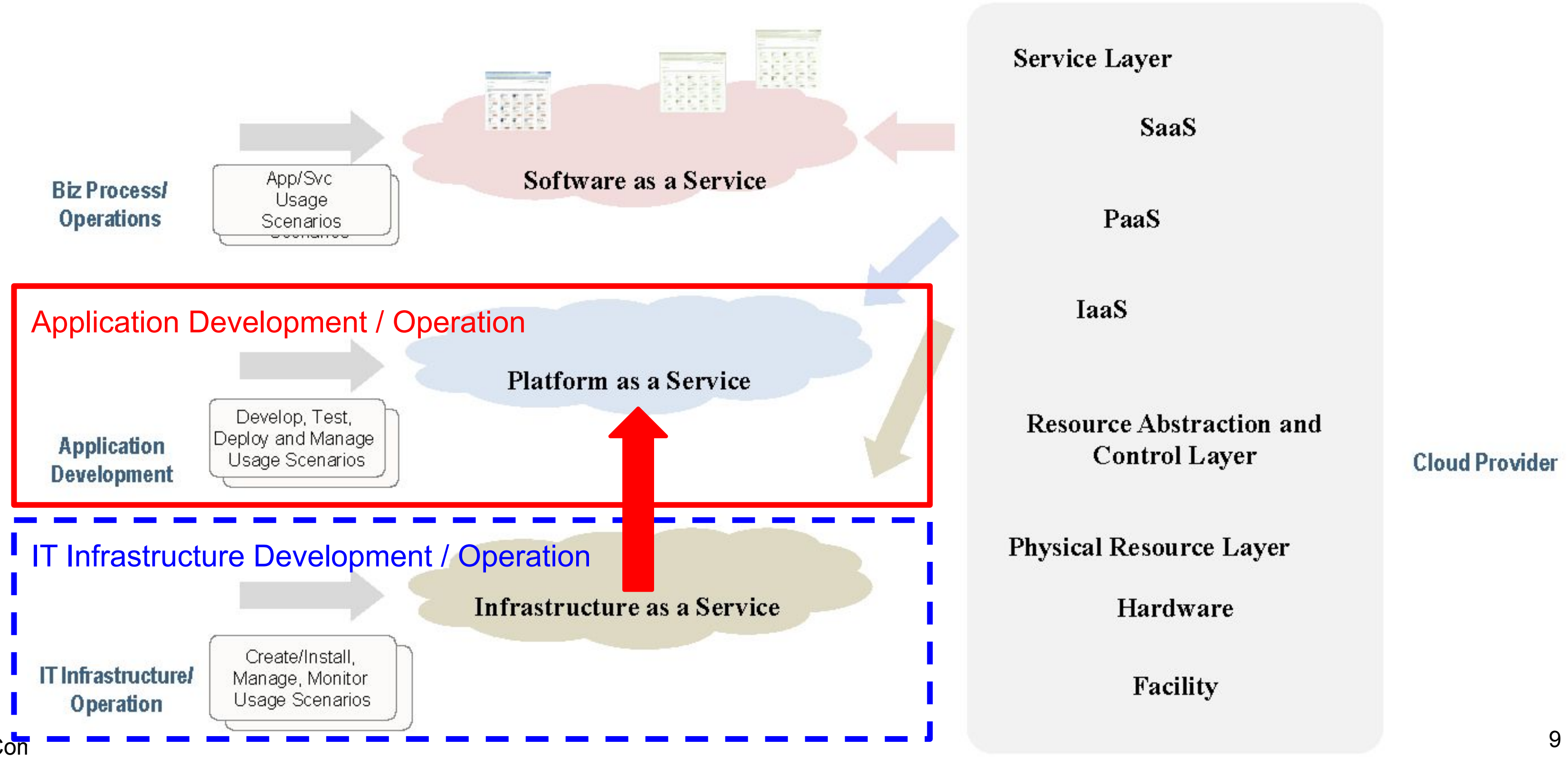
In Cloud Computing 2 (CCP2) the focus will be on:

Cloud Computing from the Developer’s & Software Architect’s perspective

or

Design, Development & Management of Applications for the Cloud

Cloud Computing Service Models


Biz Process/ Operations
App/Svc Usage Scenarios
Software as a Service
Application Development / Operation
Platform as a Service
Application Development
Develop, Test, Deploy and Manage Usage Scenarios
IT Infrastructure Development / Operation
Infrastructure as a Service
IT Infrastructure/ Operation
Create/Install, Manage, Monitor Usage Scenarios
Service Layer
SaaS
PaaS
IaaS
Resource Abstraction and Control Layer
Physical Resource Layer
Hardware
Facility
Cloud Provider

Developer care about ...

→ notes on whiteboard

Developer ought to not care about ...

- VM configuration, provisioning & monitoring
- Operating System deployment
- Driver installation & configuration
- Security patches
- Network configuration
- Disk partitioning
- Storage provisioning
- Application Runtime deployment
- Database installation & configuration
- Base Service installation & configuration
- …

Definition of PaaS

The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly configuration settings for the application-hosting environment

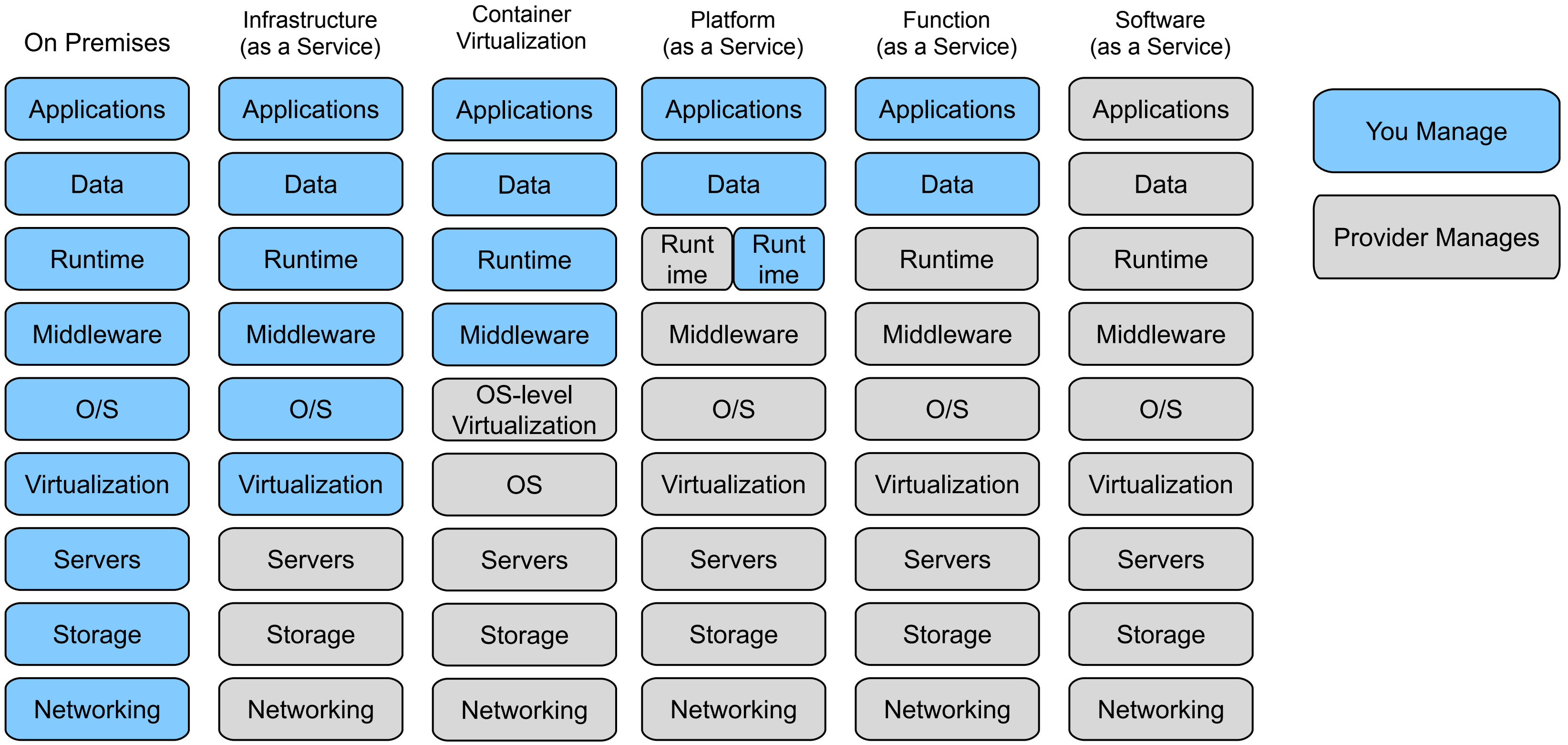
NIST

Ref: http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf

PaaS Business Motivation

- Let user focus on Business Functionality, Code and Development Process → less overhead
- Faster value to market → frequent updates & new features
- Continuous improvement → quality, user requirements
- Supports agile organization and development
- On-demand cost model (OPEX)
- Standardized Environments → reducing the number of Service Contracts / Software Silos
- Use standard services and focus on core value propositions

Cloud Computing Service Models



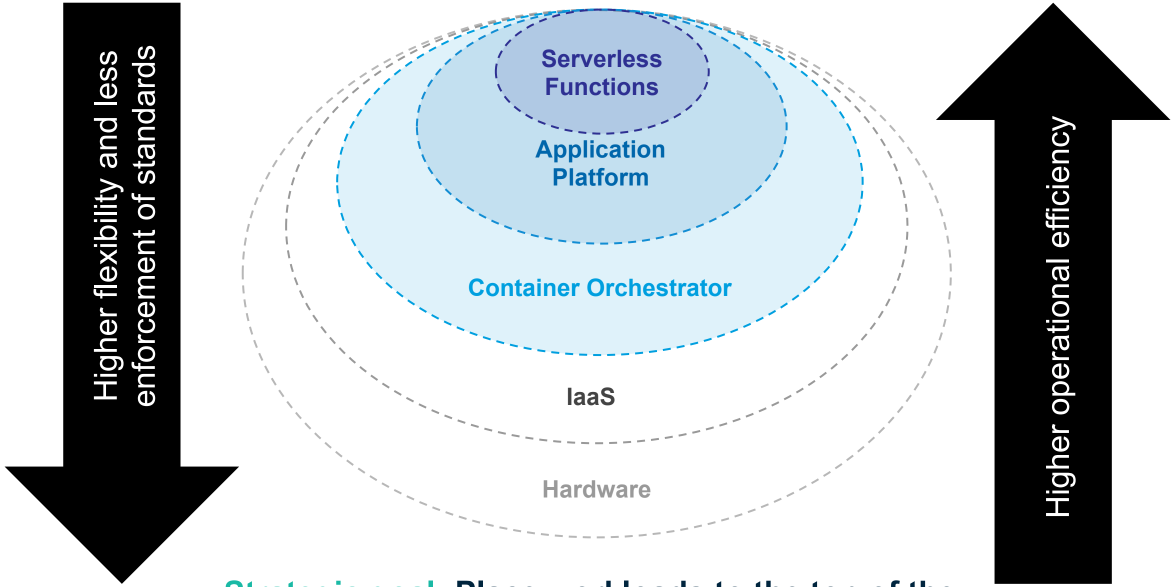
On Premises	Infrastructure (as a Service)	Container Virtualization	Platform (as a Service)	Function (as a Service)	Software (as a Service)
🟦 Applications	🟦 Applications	🟦 Applications	🟦 Applications	🟦 Applications	⬜ Applications
🟦 Data	🟦 Data	🟦 Data	🟦 Data	🟦 Data	⬜ Data
🟦 Runtime	🟦 Runtime	🟦 Runtime	⬜ Runt ime / 🟦 Runt ime	⬜ Runtime	⬜ Runtime
🟦 Middleware	🟦 Middleware	🟦 Middleware	⬜ Middleware	⬜ Middleware	⬜ Middleware
🟦 O/S	🟦 O/S	⬜ OS-level Virtualization	⬜ O/S	⬜ O/S	⬜ O/S
🟦 Virtualization	🟦 Virtualization	⬜ OS	⬜ Virtualization	⬜ Virtualization	⬜ Virtualization
🟦 Servers	⬜ Servers	⬜ Servers	⬜ Servers	⬜ Servers	⬜ Servers
🟦 Storage	⬜ Storage	⬜ Storage	⬜ Storage	⬜ Storage	⬜ Storage
🟦 Networking	⬜ Networking	⬜ Networking	⬜ Networking	⬜ Networking	⬜ Networking

🟦 You Manage

⬜ Provider Manages

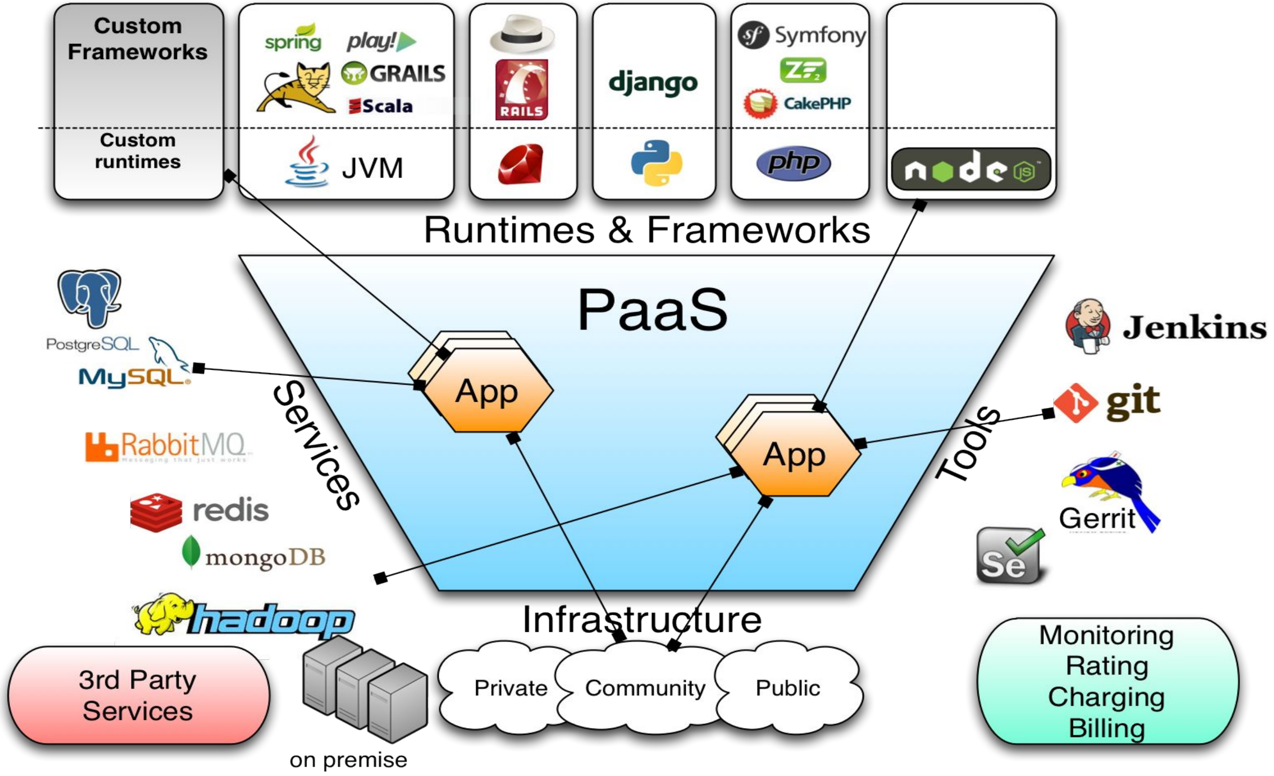
Zurich University of Applied Sciences
School of Engineering
InIT Institute of Applied Information Technology

Pick the right runtime for the right workload


Higher flexibility and less enforcement of standards
Serverless Functions
Application Platform
Container Orchestrator
IaaS
Hardware
Higher operational efficiency

Strategic goal: Place workloads to the top of the platform hierarchy

PaaS Components

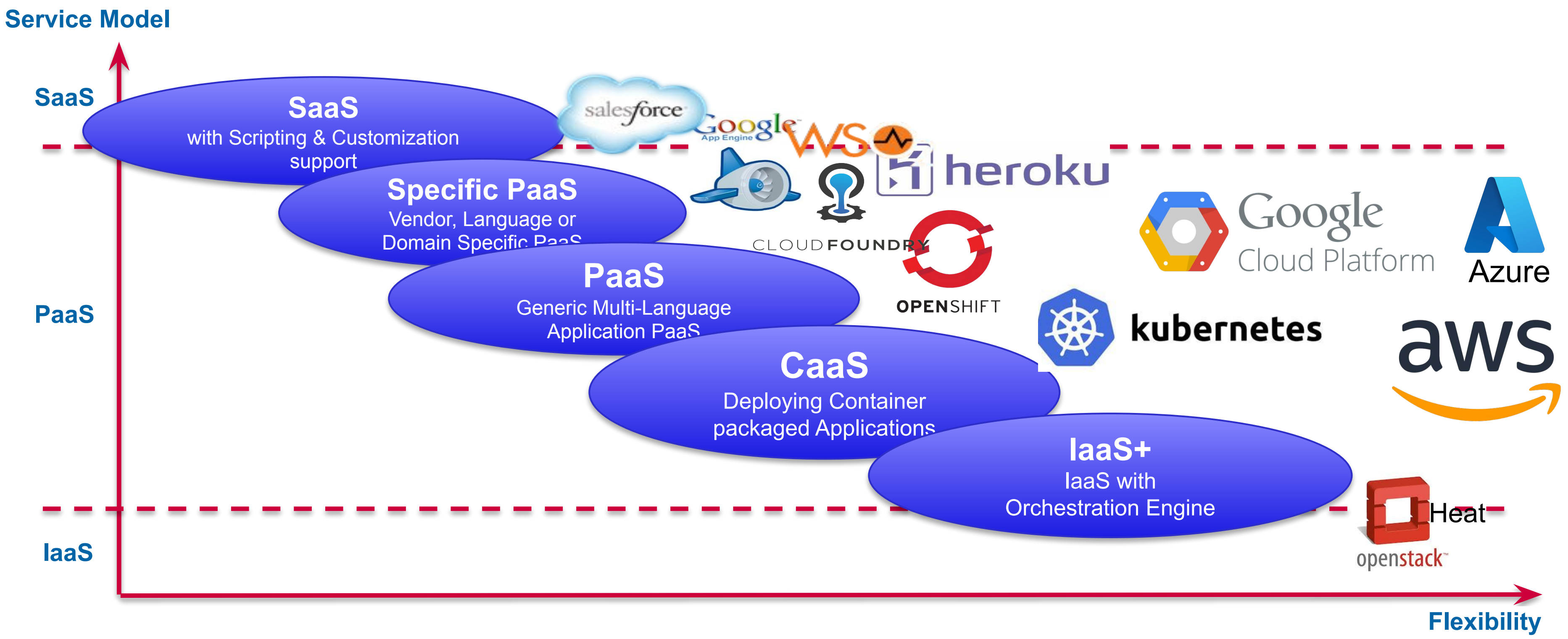


Hosting vs. IaaS+ vs. CaaS vs. PaaS

Differentiate in their delivery model

- **Hosting**
  - each tenant gets a custom arrangement of technology and all dedicated resources.
  - provides application infrastructure capabilities, but none of the characteristics of cloud computing, or standardization of technology choices, higher productivity.
- **IaaS(+)**
  - deployed on IaaS VMs. The tenant is responsible for choosing, configuring and paying for the underlying infrastructure and separately for the application platform.
  - tenant manages scaling, high availability, patching and versioning of software.
  - With some investment and coordination, this arrangement can be a legitimate cloud operation, but the customer is responsible for the arrangement.
- **CaaS**
  - Provider takes responsibility for the infrastructure to run the Container, offers rendundancy/HA features, networking, storage….
  - tenant is responsible to package application and deploy it correctly. Set-up/operate services, ...
- **PaaS:**
  - provider takes full responsibility for management and versioning of the system and application infrastructure
  - including packageing of applications
  - providing cloud characteristics of self-service, scaling, resource optimization, high availability, disaster recovery and security - all in a tenant- sensitive multi-tenant context.

Simplicity vs. Flexibility


Service Model
SaaS
PaaS
IaaS
SaaS
with Scripting & Customization support
Specific PaaS
Vendor, Language or Domain Specific PaaS
PaaS
Generic Multi-Language Application PaaS
CaaS
Deploying Container packaged Applications
IaaS+
IaaS with Orchestration Engine
salesforce
Google App Engine
WSO2
heroku
CLOUDFOUNDRY
OPENSHIFT
Google Cloud Platform
Azure
kubernetes
aws
Heat
openstack
Flexibility

Positioning PaaS vs. Containers

- **Application (PaaS / Application Runtime)**
  - you are in charge of the code and
  - apps which follow the 12-factor principles (https://12factor.net)
  - "here is my code, run it, I don't care how!"

- **Container (Container Orchestration)**
  - containerized artefacts from vendors or
  - stateful components (e.g. database clusters)
  - "here is my code bundled in a docker image, let me tell you how to run it"

Docker Swarm

PaaS Service Categories

- The PaaS Service layer can be split into categories of functionalities.
- They are usually implemented as cloud services (PaaS internal or IaaS+, SaaS) which can be accessed through APIs.
- Often difficult to distinguish if it is PaaS or SaaS.
- Service Categories often get their dedicated …PaaS or ...aaS acronyms, summarized as xPaaS or XaaS
- The following slides give a brief overview of the main PaaS Service Categories

Application Platform Services - aPaaS

- Provides core PaaS functionality to deploy, run and manage applications
- Supports multiple Language Runtimes and Programming Frameworks
- Combined with tools for development, management, composition and typically data persistence
- Two types of cloud-enabled Application Platform Software exists
  - PaaS Frameworks: Designed to host or manage multiple applications
    - Pivotal CF, Red Hat OpenShift Dedicated and Online, Google App Engine, Apprenda, WSO2/Apache Stratos
  - Cloud-enabled Application Platforms: Not really PaaS, but more environments for composing distributed applications
    - Mendix, Progress Software Rollbase, MIOsoft MIOceap, Software AG AgileApps Live

Other Categories - xPaaS

- Cloud Integration Platform Services (**iPaaS**)
  - offers enterprise integration capabilities: e.g. enterprise service buses (ESBs), data integration tools, B2B gateways, message-oriented middleware, managed file transfer and API management.
- Cloud DBMS and Data Store Services (**dbPaaS**)
  - DBMSs or data stores engineered to run as a scalable, elastic, multitenant service available from cloud providers.
- Cloud Business Process Management Services (**bpmPaaS**). BPM platform capabilities as Service. It minimally includes:
  - A graphical business process and/or rule modeling capability
  - A process registry/repository to handle the modeling metadata
  - A process execution, and either a state management engine and/or a rule engine
- Cloud Business Analytics Services (**baPaaS**)
- Cloud Application Development Life Cycle Management Services (**ADLM PaaS**)
- Cloud Mobile Back-End Services (**MBaaS**)
- Cloud In-Memory DataGrid Services (**IMDG Services**)

Domain Specific PaaS

PaaS for vertical markets, making domain-specific business processes available in the form of services.

This might catch on in two ways:

- Enterprises / Start-ups will use it to help development teams build new apps fast with less involvement from IT
- Work closely with business partners in fleshing out services-based ecosystems.

Examples:

- PaaS for Mobile Applications
- PaaS for IoT
- PaaS for Robotics - ICCLab Cloud Robotics Research, https://www.youtube.com/watch?v=NCjp2txUSmc
- PaaS for Healthcare
- PaaS for DataAnalytics / Scientific Computing

Market Overview

PaaS providers & products (software, services mixed)

Standardized /OpenSource offering (based on OpenSource o.a., low Lockin)

- IBM Bluemix
- Hewlett Packard Enterprise
- Pivotal
- swisscom
- canopy the atos cloud
- HUAWEI
- FUJITSU
- anynines
- SAP HCP (v2)
- CloudFoundry
- TECTONIC by CoreOS
- redhat OpenShift
- KISMATIC
- Kubernetes
- MESOSPHERE Mesos
- Engine Yard DEIS

Proprietary services & products (high lock-in, not open source)

- Microsoft Azure
- Google App Engine
- heroku
- amazon Beanstalk
- ORACLE
- apprenda
- Engine Yard
- FLOW APP ENGINE
- SAP
- APCERA

Google App Engine

One of the first PaaS offerings: introduced in 2008

- Originally with Python support only
- Meanwhile:
  - Go, Java, .NET, Node.js, PHP, Python, or Ruby, using pre-configured runtimes
  - Support for custom runtimes to write code in any language
- Google App Engine manages app hosting, scaling, monitoring, and infrastructure
- Connects with Google Cloud storage products, such as
  - Cloud SQL, Firestore in Datastore mode, and Cloud Storage.
- You can also
  - Connect to managed Redis databases
  - Host third-party databases such as MongoDB and Cassandra on Compute Engine, another cloud provider, on-premises, or with a third-party vendor.
- Features Web Security Scanner to identify security vulnerabilities as a complement to your existing secure design and development processes.

App Engine

Part of

Google Cloud

Amazon Elastic Beanstalk

Based on Amazon EC2 (Deploying services in EC2 VMs), Introduced 2010

Support for

- PHP
- Java/Tomcat/Node.js
- Python, Ruby

Access to AWS Services:

- Amazon RDS (MySQL and Oracle)
- Amazon SimpleDB
- Amazon DynamoDB

AWS Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications and services developed with Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker on familiar servers such as Apache, Nginx, Passenger, and IIS.

You can simply upload your code and Elastic Beanstalk automatically handles the deployment, from capacity provisioning, load balancing, auto-scaling to application health monitoring. At the same time, you retain full control over the AWS resources powering your application and can access the underlying resources at any time.

There is no additional charge for Elastic Beanstalk - you pay only for the AWS resources needed to store and run your applications.

Heroku

- Independently founded in 2007, now owned by Salesforce
- Initially **Ruby on Rails** centric
- Introduced **Java and Node.js, Python** support in 2011
- Highly flexible public cloud, many service options (https://addons.heroku.com)
  - PostgreSQL (standard store)
  - Amazon RDS
  - MySQL
  - Redis
  - MongoDB
  - RabbitMQ
- Invented the concept of **buildpacks** (structured deployment scripts, user providable)
- Running on AWS.

"Heroku is a platform as a service based on a managed container system, with integrated data services and a powerful ecosystem, for deploying and running modern apps. The Heroku developer experience is an app-centric approach for software delivery, integrated with today's most popular developer tools and workflows."
https://www.heroku.com/platform

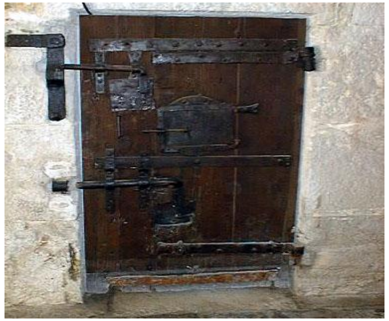
… others (all Closed Source)

Microsoft Azure

- Started as PaaS with .Net (now also Java, Python, Ruby, …)
- Added IaaS services 2012 (also Unix images)

Closed Source

- Threat: Lock-In
- Only single provider
- Proprietary Services & APIs
- Difficult to move Apps (need to rewrite parts)



Open Source PaaS Frameworks

redhat

OPENSHIFT

CLOUD FOUNDRY FOUNDATION

CLOUD FOUNDRY

IBM Bluemix™

3 Flavors: Online, Dedicated, Origin

VMware Tanzu

swisscom Enterprise Application Cloud

SAP® SAP CLOUD PLATTFORM

SUSE

- Similar Functionality, Different Implementation (in details)
- OpenShift tailored to RedHat-like platforms
- CloudFoundry Ubuntu, CentOS, Windows → more OS agnostic

Recap: Pro and Con of PaaS Adoption

- See whiteboard