Software as a Service (SaaS)

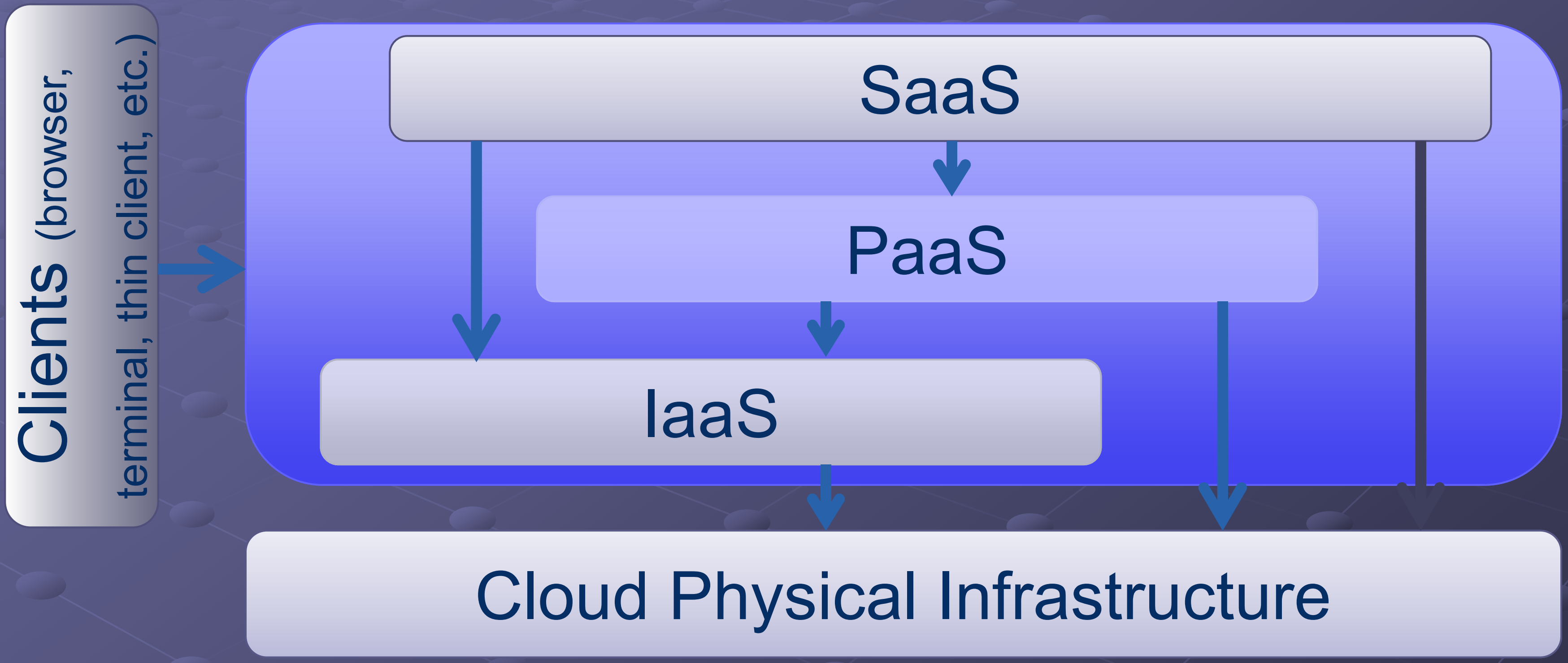
Prof. Tahar Kechadi

School of Computer Science & Informatics

Outline

- Define SaaS
- Pros and Cons
- Case studies

Keep the hierarchy in mind!



SaaS: An Overview

- Software as a Service (SaaS) is the model in which an application is hosted as a service to customers who access it via the Internet
  - The customer does not have to maintain it or support it

The software is used out of the box and there is no need for changes or integration to in-house systems

Why SaaS?

Users who are not inclined to perform software development but have need of high-powered applications can benefit from SaaS. Some of these applications include:

- Customer resource management (CRM)
- Video conferencing
- IT service management
- Accounting
- Web analytics
- Web content management

History of SaaS

- During late 90s: Software applications would always be installed on to the same machines on which they were going to be run on
  - Internet connections were so slow
  - SaaS: very expensive solution
- XXI century :
  - Improvements in Internet speeds and an increase in availability: SaaS implemented cheaply, work efficiently, without any lag or time delays
  - Different perspectives:
    - software vendors: offering software services to consumers using a subscription-based model
    - Consumers: by using a subscription-based model, they would not have to pay large amounts of money upfront and had the ability to only pay for the services that they required.

Objectives of SaaS

- To make the management and control of software easier
- To take the management strain away from consumers
- To make software services available globally
- To provide a single instance of a software service to multiple users
- To create flexible payment models for software services

What SaaS is and what it is not

- SaaS is not Software + Service
- Do not have to install any application onto their machines
  - certain features are currently just too difficult to implement across the Internet or run within web browsers efficiently
- Users cannot work offline
- Privacy concerns
- Greater Customisability

Multitenant of SaaS Solutions

- Two or more clients may share the same server resources
- Share database resource:
  - Depending on size, fees, etc.
- Multitenant solution may be difficult, expensive or impossible.

Service-oriented Architecture (SOA)

- Application development methodology
- Integrating one or more web services
  - Web services are solutions that programs can call across the web to perform specific tasks.
- A set of web services: API
- SaaS application interacts with a user – a web service interacts with a program.

Mashup

- Collection of services joined to create an overall solution.
- Web-based:
  - User’s browser combines the various content sources to create a unified display
- Server-based:
  - An application running on a server combines the data

OpenSaaS

- SaaS solution:
  - Use a specific programming language
  - Run on a specific OS
  - Use a specific DBMS
- OpenSaaS:
  - Use a opensource programming language
  - Run on an open source OS and DBMS
  - Move data to different applications

Pros of SaaS

- Reduce or eliminate the need for an on-site data centre
- Eliminate the need for application administration
- Allow customers to pay on demand for software use, normally on a per-user basis
- Scalability: application, processor and data storage
- Device independent access to applications
- Increase disaster recovery and business continuity

Example: Microsoft Office 365

- Microsoft Office vs. Open Office
- Office 365:
  - Pay-by-the-month subscription to Office apps
  - Access, edit documents from any computers
  - Collaborate and share documents easily

Delivering software: traditional

- Software vendor
  - Software customized for platforms/customers
    - software: $4000/user, support: $800/user/year
    - Long and expensive customization
    - A department is necessary to distribute software
    - Slow to iterate new versions
  - Success story: Oracle (2009 $12b)
- Customer
  - Particular hardware/software platform
  - IT specialists to manage the system
    - extra cost: $1300/user/year

Delivering software: open-source

- Software vendor
  - Low development cost
    High (individual) maintenance cost
    - software: $0, support: $1600/user
  - Difficult to monetize support (gold, platinum, …)
  - On-demand solutions
  - Success story: Red Hat
- Customer
  - Hardware/software platform can be basic
  - IT staff still needed

Delivering software: outsourcing

- Software vendor
  - Traditional development/maintenance
    - software: $4000/user, support: $800/user/year
- Customer
  - Hardware/software platform still needed
  - IT outsourced to a third party
    - service: <$1300/user/month
  - Outsourcers manages software @client or @home
  - Success story: Infosys

Delivering software: hybrid

- Software vendor
  - Massive efficient software maintenance
    - software: $4000/user, support: $800/user/year
      service: $150/user/month
  - Software is managed @client or @home
  - Success story: Callidus Software
- Customer
  - Hardware/software platform still needed
  - IT outsourced to the software vendor

Delivering software: SaaS

- Software vendor
  - Infrastructure is managed by the vendor and/or outsourced to IaaS providers
  - Platform is managed by the vendor and/or outsourced to PaaS providers
  - Software is managed by the vendor
    - <$100/user/month
  - Mutli-tenant architecure
- Customer
  - Internet access

Summary

- Describe SaaS
- SaaS techniques
- Pros and Cons
- Example