Deusto tech
Tecnológico Fundación Deusto
Teknologikoa Deustu Fundazioa
Deusto Technology Foundation

Towards Future Internet: Web 3.0, Internet of Services & Internet of Things

8 de Julio 2009, 11:30-13:30
Sala de Videoconferencias, ESIDE

Dr. Diego López de Ipiña (y algunos de mis doctorandos)
dipina@eside.deusto.es

Agenda

- **Future Internet**
  - Motivation
  - Definition
  - Architecture: pillars and foundation
    - Internet by and for People
    - Internet of Contents and Knowledge
    - Internet of Things
    - Internet of Services
  - Research challenges

Agenda

- **Future Web**
  - Web Evolution: Web 3.0
  - Future Browsers
  - Future Search
  - Web as a Platform
    - Client-side: RIA, HTML 5.0
    - Server-side: WOA, REST, Cloud Computing

Agenda

- **Semantic Web**
  - Semantic Web
  - Lower-s Semantic Web
  - Semantic Mash-ups

Why do we need a New Internet? (1)

- The current Internet, with over 1.5 billion users worldwide, is a great success in terms of connecting people and communities, BUT:
  - It was **designed in the 1970s** for purposes quite unlike today's heterogeneous application needs and user expectations
- The **current Internet has grown beyond its original expectation**s **and** beyond its original **design objectives**.
  - Though the Internet infrastructure has evolved with changing applications, its underlying architecture has to date slowly evolved
    - This underlying architecture was not created to function as a global critical infrastructure, and it has a number of fundamental limitations.
  - **Progressively reaches a set of fundamental technological limits** and is impacted by operational limitations imposed by its architecture
    - It 'only just' works - whereas tomorrow's applications will attract more users to new applications needing greater mobility, security, wider bandwidth, reliability, and enhanced interactivity

Why do we need a New Internet? (2)

- **Many partial solutions have been progressively developed and deployed** to allow the Internet to cope with the increasing demand in terms of user connectivity and capacity.
  - **A growing consensus** among the scientific and technical community that the methodology of continuously **"patching" the Internet technology will not be able to sustain its continuing growth and cope with it at an acceptable cost and speed**
- The **current Internet architecture is progressively reaching a saturation point in meeting increasing user's expectations and behaviors as well as progressively showing inability to efficiently respond to new technological challenges**
  - In terms of security, scalability, mobility, availability, and manageability, but also socio-economical challenges.

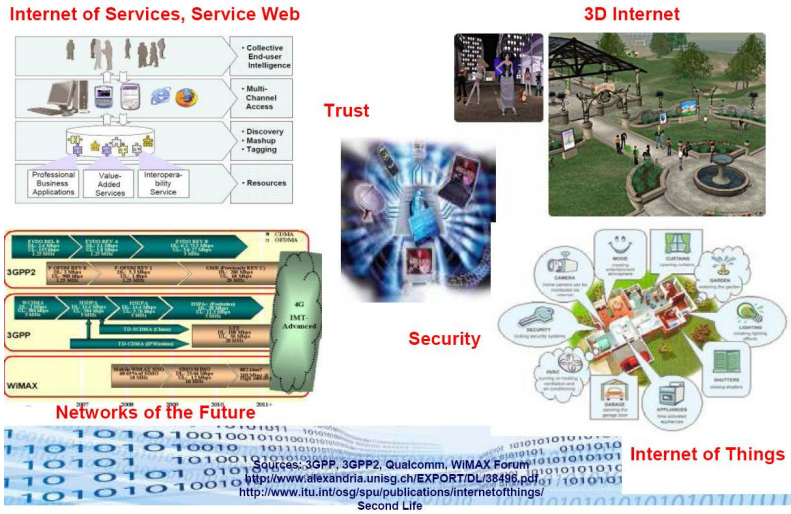
What is the Future Internet?

- Buzzword which summarizes **efforts to progress towards a better Internet**, either through:
  - Small, incremental evolutionary steps or
  - Complete redesigns (clean slate) and architecture principles
- Key players:
  - **Global Environment for Network Innovations** (**GENI**): http://www.geni.net/
    - More based on networks
  - **AKARI** project in Japan, http://akari-project.nict.go.jp/eng
  - **Future Internet** – http://www.future-internet.eu/

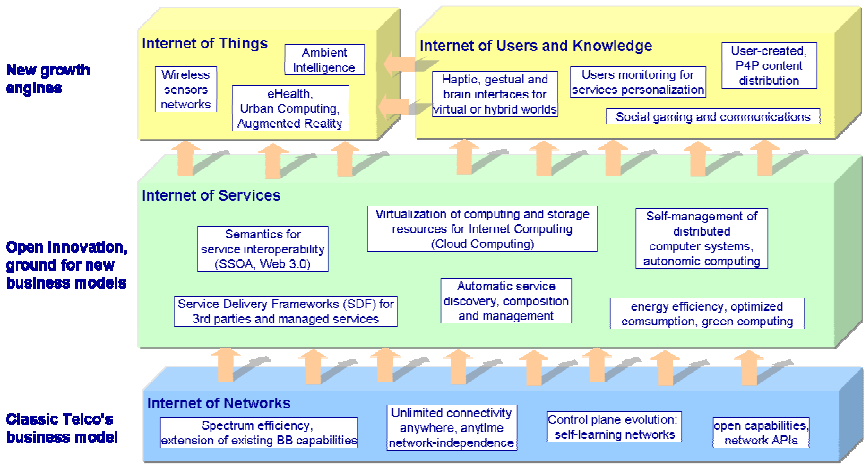
Future Internet (FI) Aim

- It should **offer** all users **a secure, efficient, trusted and reliable environment**, that:
  - Should **allow open, dynamic and decentralized access to the network connectivity service and information**
  - Be **scalable, flexible and adapt its performance to the user needs and context**.

Future Internet Vision



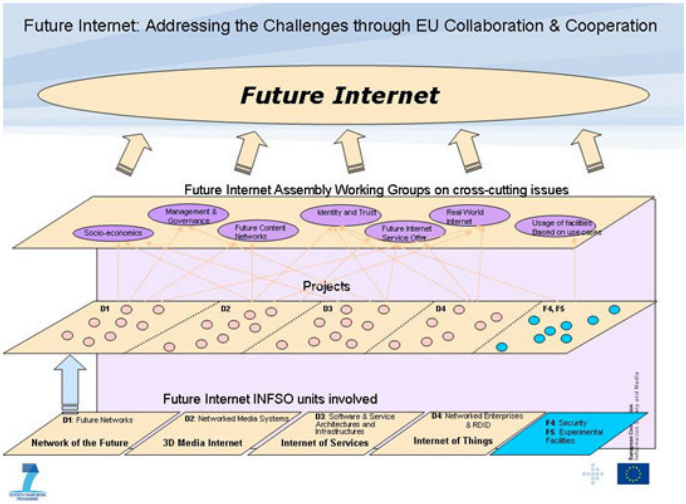
Future Internet Architecture



The BLED Declaration:
Towards a European approach to the Future Internet

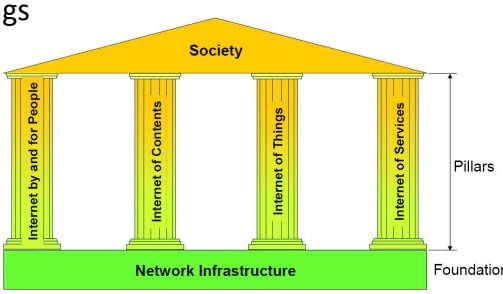
- "A significant change is required and the **European** Internet scientific and economic actors, researchers, industrialists, SMEs, users, service and content providers, now assert the **urgent necessity to redesign the Internet, taking a broad multidisciplinary approach, to meet Europe's societal and commercial ambitions"**
  - http://www.future-internet.eu/fileadmin/documents/bled_documents/Bled_declaration.pdf

Future Internet in 7th FP



Pillars and Foundation of Future Internet

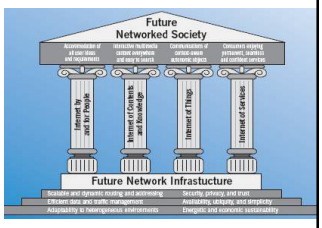
- The 4 pillars of Future Internet rely on the Future Internet Networking Infrastructure foundation:
  - Internet by and for People
  - Internet of Contents and Knowledge
  - Internet of Services
  - Internet of Things



Deusto tech
Tecnológico Fundación Deusto
Teknologikoa Deustu Fundazioa
Deusto Technology Foundation

Universidad de Deusto
Deustuko Unibertsitatea

Pillars and Foundation of Future Internet



- All elements of the Future Internet (foundation and pillars) need each other and are mutually dependent
- **New services and applications** are prerequisite for investments in new infrastructure, since Infrastructure without necessary capabilities can not support new services and applications
  - **Technology pull**
- **New infrastructure technologies open new opportunities for new services and applications**
  - **Technology push**
- Therefore, cooperation between all stakeholders required for a successful Future Internet

Deusto tech
Tecnológico Fundación Deusto
Teknologikoa Deustu Fundazioa
Deusto Technology Foundation

14

Universidad de Deusto
Deustuko Unibertsitatea

Internet by and for People

- A new concept of Internet fostering:
  - **User experience**: development of new ergonomic user interfaces based on advanced multi-modal interaction models.
  - **Active users**: new tools to allow final users to create and share personalised services (not only contents, but also applications).
    - The emergence of **prosumers**: people/communities will be part of the creative flow of content and process, and not just consumers.
  - **Content and user awareness**: content and service personalisation according to context and personal preferences.
  - **Knowledge of users**: extract user information, monitoring their behaviour without hindering their privacy
    - Web 3.0, semantic technologies, knowledge exchange, processing and generation by machines are important in this area

Internet Contents and Knowledge

- Some of the innovation areas are:
  - **New User Devices and Terminals**: advanced multimedia devices with much functionality integrated, scalable, auto-configurable, open programming APIs and so on.
  - **Social media applications**: software and tools to integrate multimedia contents oriented towards social relationships and communications.
  - **Digital media**: tools opened to create contents by the very user and their distribution over increasing number of locations.
  - **Cognitive intelligence**: web evolution to Web 3.0 will enable advanced multimedia search engines.
  - **Semantically tagged media**: is the foundation by which applications not only provide information but also intelligently process information.

Internet of Things

- A **world-wide network of uniquely addressable and interconnected objects, based on standard communication protocols**.
  - IoT will exhibit a much **higher level of heterogeneity**, as objects totally different in terms of functionality, technology and application fields will belong to the same communication environment.
  - Novel protocols based on the semantic of the language must be developed, if the IoT has to scale to the zillions of objects around us
  - The combination of **RFID and sensors enables a cost-effective and robust system of item identification and context awareness**, changing the current Internet usage completely from a request-and retrieve to a **push-and-process paradigm**.

Internet of Things

- Some issues that will have to be addressed are:
  - Sensor networks will require capacity to process big amounts of information coming from “connected things” and so offer more useful services.
  - Ethical and societal dimension: maintain the balance among personalisation, privacy and security.

Application Evolution of RFID

* electronic Product Code

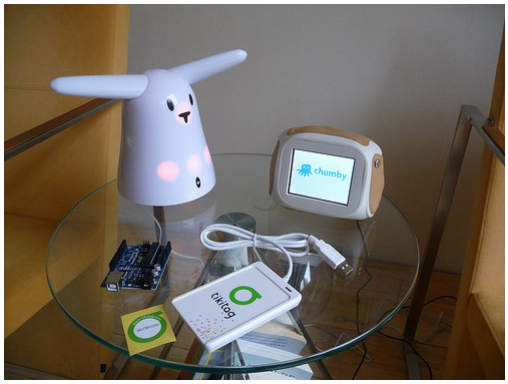
Sources: Siemens CT / STA / CD S, Mitsubishi Electric Research

Deusto tech
Tecnológico Fundación Deusto
Teknologikoa Deustu Fundazioa
Deusto Technology Foundation

Universidad de Deusto
Deustuko Unibertsitatea

Is IoT already a market opportunity?

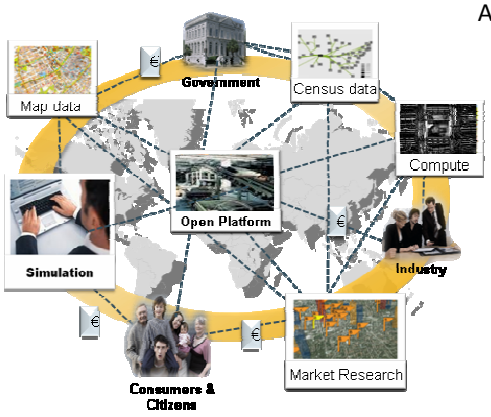
- Arduino, tiki tag, nabaztag, chumby



Deusto tech
Tecnológico Fundación Deusto
Teknologikoa Deustu Fundazioa
Deusto Technology Foundation

Universidad de Deusto
Deustuko Unibertsitatea

Internet of Services

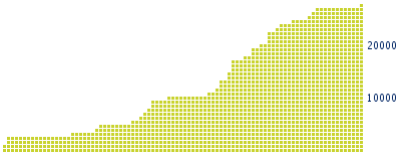


Adapted from SAP Research, 2008, and SEEKDA, 2008

A multitude of connected IT services, which are offered, bought, sold, used, repurposed, and composed by a worldwide network of service providers, consumers, aggregators, and brokers

- resulting in -

a new way of offering, using, and organising IT supported functionality

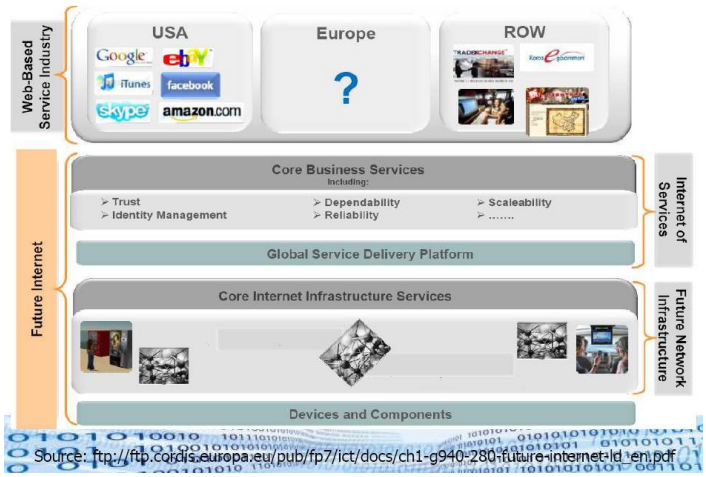


Number of Web services found by SEEKDA crawler during the past 25 months

Internet of Services

- Umbrella term to describe several interacting phenomena that will shape the **future of how services are provided and operated on the Internet**
  - http://services.future-internet.eu/index.php/Main_Page
- Three major domains of development are:
  - **Internet-scale service oriented computing** – a service-oriented Internet should allow access to complex physical computing resources, data or software functionality in the form of services
  - **Contextualized, proactive, and personalized access to services:**
    - Context-awareness – interaction personalized to context
    - Seamless multi-modal interaction
    - End-user empowerment – it will be more easy for users to design their own orchestration of services as well as to configure their own service front-end web access to services by means of self-servicing and mashing up service front-end resources published in catalogues/stores available on the Internet
    - Collaborative – prosumers, tagging
  - **Service orchestration and the rise of core services** – several layers of services, from fundamental infrastructure services – like those provided by clouds to specific, data-, information-, application-like and user interfacing services.

Internet of Services Architecture

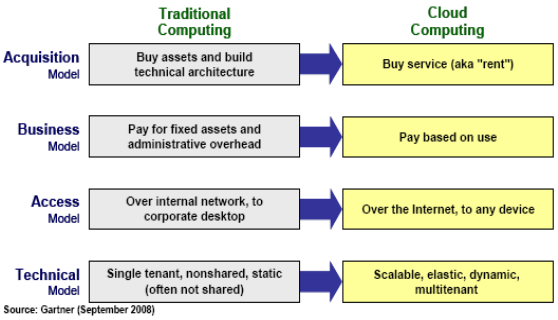


Internet of Services

- **Some interesting application areas of Internet of Services**
  - **Cloud computing**: service virtualization and resource optimization both in processing and storage capacity
  - **Green IT**: energy consumption optimization (efficiency).
  - **Open service platforms**: service modularization so that end-users can easily integrate them
  - **Autonomic computing**: self-managing systems

Virtualised Infrastructures e.g. Cloud Computing

An emerging computing paradigm where data and services reside in massively scalable data centers and can be ubiquitously accessed from any connected device over the Internet[1]

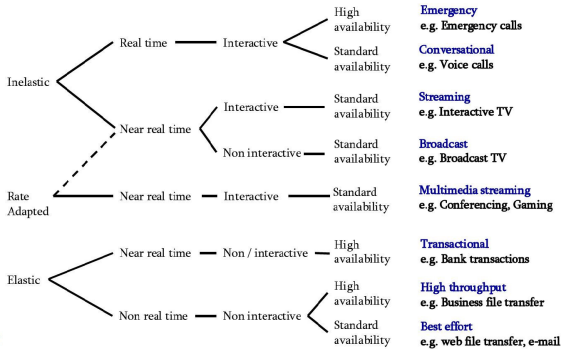


Merrill Lynch:
Cloud computing market opportunity by 2011 = $95bn in business and productivity apps + $65bn in online advertising = $160bn

(1) Source: IBM

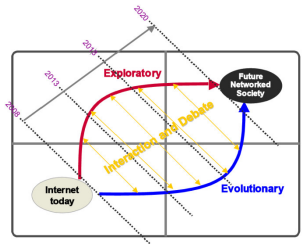
Infrastructure Foundation

- Network infrastructure **supports the pillars and sustains the resulting capacity and performance requirements that Future Internet** will provide
- The main drivers here are the **use of the Internet as a common infrastructure for interconnecting more than computing machines** e.g. sensor networks, mobile devices, wearables, M2M, RFID and so on .



Evolutionary / Revolutionary (Clean-Slate) Approaches

- Two approaches to address challenges of Future Internet:
  - **Evolutionary** – builds on the evolution of the current existing Internet to conceive pragmatic and viable solutions for commercial rollout
  - **Revolutionary** – starts from a clean slate to eliminate legacy Internet design constraints
    - There is a need to separate clean slate research from clean slate deployment
      - Clean-state research results are expected to feed the evolution of the Internet
- Both approaches target the same usage vision and will have to be synchronized



Key Generic Challenges

1. Routing and addressing scalability and dynamics
2. Resource (forwarding, processing, and storage) and data/traffic manageability and diagnosability
3. Security, privacy, trust, and accountability
4. Availability, ubiquity, and simplicity
5. Adaptability and evolvability to heterogeneous environments, content, context/situation, and application needs (vehicular, ambient/domestic, industrial, etc.)
6. Operating system, application and host mobility /nomadicity
7. Energetic and economic sustainability
8. Conflicting interests and dissimilar utility
9. Searchability/localisation, selection, composition, and adaptation
10. Beyond digital communication: semantic (intelligibility of things and content, language, etc.), haptic, emotion, etc

Key Technological Challenges

- **Internet by and for People**: accommodate anticipated and unanticipated people and community expectations together with their continuous empowerment, cultural acumen, and self-arbitration (by recognizing that access and use of information as well as associated processing means are common non-discriminatory universal rights).
- **Internet of Contents and Knowledge**: access by advanced search means and interact with multimedia content (e.g. 3D and virtual reality) that can be created, and manipulated by professionals and non-professionals and be distributed and shared everywhere on any terminal per needs.
- **Internet of Things**: context-aware autonomic objects able to generate automatic code and human-controlled behaviors, exploiting peer-to-peer bio-inspired communication models.
- **Internet of Services**: service "consumers" look for the perfect interactivity in context. With “perfect” we mean here **permanent** (i.e. interactivity that has no time limits), **direct** (i.e. the service consumer is only concentrated on the benefits of the service he/she is using), **seamless** (i.e. the interaction is performed using the “typical” devices of the context), and **confident** .

Network Foundation Challenges

- As a result of the Internet growth and the increasing communication requirements, many patch solutions have been progressively developed and deployed to enable the Internet to cope with the increasing demand in terms of user connectivity and capacity.
  - The current methodology of **“patching” the Internet technology will not be able to sustain its continuing growth** and cope with it at an acceptable cost and speed.
    - Security, privacy and trust
    - Accountability
    - Manageability and Diagnosability
    - Availability (maintainability and reliability)
    - Scalability
    - Mobility
    - Heterogeneity (devices, environments, applications)

Internet by and for People Challenges

- **Increase the knowledge of the user**, learning their habits and needs to better design future applications, interfaces and services
  - **Knowledge of users**: services and the web should be adaptable by and accustomed for the user.
  - **Content and user awareness**. This area is focused on recommendation systems, particularly for mobile web and localized services
  - **Active users**. There are evidences that the tendency towards more and better "free services" will be growing over time. Also, the evolution of users towards "prosumers", result in fundamental re-definition of content creation

- **User experience**: new ergonomic interfaces and advanced interaction mechanisms including multimodality & semantic combination and adaptation of information from different sources

Internet of Contents and Knowledge Challenges

- Relates to the generation and processing of content and the transformation of that content into useful information. It also includes the aspects regarding the user and its characterization and relationships between user and content:
  - **Digital Content** – 'Content' refers to the 'understandable information made available to a user at any stage of the value chain', including both the 'essence' – the data – as well as the metadata that describes the essence and allows it to be searched, routed, processed, selected, and consumed.
  - **Distributed Media Applications** – prosumers will play a leading role and allow for an automated selection without need to care anymore how recipients are going to access the produced content.
  - **New User Devices and Terminals** - Users should be able to access services wherever they are, whatever terminal they are using, with seamless handover as they change from one terminal to another.

Internet of Things Challenges

- **Handle the large amount of information** coming from the things and to combine it to give useful services.
- The current network structure is not suited for this exponential traffic growth, there is a need by all the actors to **re-think current networking and storage architectures**.
- Imperative to **find novel ways and mechanisms to find, fetch, and transmit data.**
  - Distributed, loosely coupled, ad-hoc peer-to-peer architectures connecting smart devices might represent the network of the future.
- Some specific challenges:
  - **Discovery** of sensor data — in time and space
  - **Communication** of sensor data: Complex Queries (synchronous), Publish/Subscribe (asynchronous)
  - **Processing** of great variety of sensor data streams
    - In-network processing of sensor data: correlation, aggregation, filtering

Internet of Services Challenges

- **Allow access to services based on** technical characteristics such as IP-location or web service identifiers but also based on **contextual information** (e.g. using geographical context or business context).
- **Services** can be **searched, identified and composed into business process components**.
- Service consumers look for the "**Perfect interactivity**":
  - **Permanent** (i.e. an interactivity that has no time limits)
  - **Transparent** (i.e. the service consumer is only concentrated on the benefits of the service he/she is using)
  - **Seamless** (i.e., supporting mobility of users across different devices without interruption),
  - **Context-aware** (i.e. the interaction gets adapted to context in its widest sense, including characteristics of devices, location, user preferences or social networks the user belong to),
  - **Empowering** (i.e., users are enable to self-configure the way they want to get access to services) and
  - **Trustworthy** (i.e., users feel confident that their interaction with services is safe)
- Broad range of emerging applications:
  - Cloud computing (SLA, billing), open service platforms, autonomic computing, green IT

Functional Properties of FI

- Accountability
- Security
- Privacy
- Availability (maintainability and reliability)
- Manageability, and diagnosability (root cause detection and analysis)
- Mobility, and nomadity
- Accessibility
- Openness
- Transparency (the end-user/application is only concerned with the end-to-end service, in the current Internet this service is the connectivity)
- Neutrality

Architectural Properties of FI

- Distributed, automated, and autonomy (organic deployment)
- Scalability (e.g. routing scalability -> log(n) where n is the number of nodes and computational scalability i.e. to allow support of any business size)
- Resiliency and survivability
- Robustness/stability
- Genericity (e.g. support multiple traffic (streams, messages, etc.), independent of infrastructure partitioning/divisions, device/system independent)
- Flexibility (e.g. support multiple socio-economic models, and operational models)
- Simplicity
- Evolvability: evolutionability and extendability
- Heterogeneity (e.g. wireline and wireless access technologies,)
- Carbon neutrality

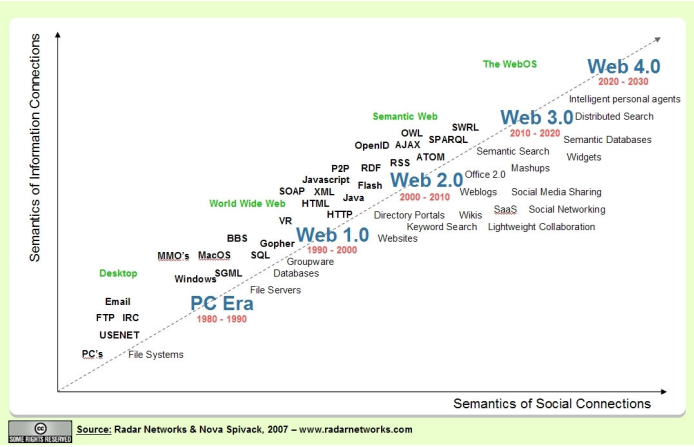
7FP Projects Addressing FI Challenges

- Detailed listing of current ongoing projects on the Future Internet topic:
  - http://www.future-internet.eu/activities/fp7-projects.html

Future Web

- **Semantic Web:**
  - Machines have started to understand the content they've been hosting and can now obtain information from the semantic of the documents.
    - Technologies: XML, RDF, SPARQL, OWL, µFormats...
- **Ubiquitous Web:**
  - Provides a framework for exposing device coordination capabilities to Web applications.
  - Seeks to broaden the capabilities of browsers to enable new kinds of web applications, particularly those involving coordination with other devices.
    - Connecting a camera phone to a nearby printer, using a mobile phone to give a business presentation with a wireless projector ...
- **Web 3.0 and beyond!**

Web Evolution



Web Evolution

- According Nova Spivack, the **development of the Web moves in 10-year cycles**:
  - In the Web's **first decade**, most of the **development focused on the back end, or infrastructure**, of the Web.
    - Programmers created the **protocols** and code languages we use to make Web pages.
  - In the **second decade**, **focus shifted to the front end and the era of Web 2.0 began**.
    - Now people use Web pages as platforms for other applications.
    - They also create mashups and experiment with ways to make Web experiences more interactive.
    - We're at the end of the Web 2.0 cycle now.
  - The **next cycle will be Web 3.0, and the focus will shift back to the back end**.
    - Programmers will refine the Internet's infrastructure to support the advanced capabilities of Web 3.0 browsers.
  - Once that phase ends, we'll enter the era of **Web 4.0**.
    - Focus will return to the front end, and we'll see thousands of new programs that use Web 3.0 as a foundation
- Source: http://www.howstuffworks.com/framed.htm?parent=web-30.htm&url=http://www.intentblog.com/archives/2007/02/nova_spivack_th.html

Web 3.0

"People keep asking what Web 3.0 is. I think maybe when you've got an overlay of scalable vector graphics - everything rippling and folding and looking misty - on Web 2.0 and access to a semantic Web integrated across a huge space of data, you'll have access to an unbelievable data resource."

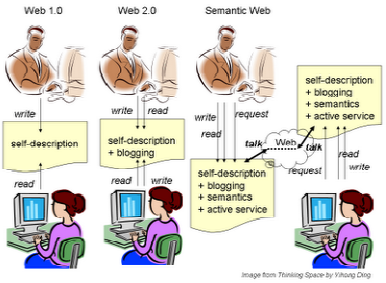
Tim Berners-Lee, 2006

*"**Web 3.0**, a phrase coined by John Markoff of the New York Times in 2006, refers to a supposed third generation of Internet-based services that collectively comprise what might be called "the intelligent Web" -- such as those using semantic web, microformats, natural language search, data-mining, machine learning, recommendation agents, and artificial intelligence technologies - which emphasize **machine-facilitated understanding of information in order to provide a more productive and intuitive user experience**."*

– http://www.intentblog.com/archives/2007/02/nova_spivack_th.html

Web 3.0

- **Web 3.0** is a mixture of different technologies that change, in a very impressive way, **how users interact with the web and how the web interacts with the physical world**.
  - Semantic Web
  - Ubiquitous Web
  - 3D Interfaces
  - Artificial Intelligence
  - …



Web 3.0: Convergence of Trends

- **Ubiquitous Connectivity**
  - Broadband adoption
  - Mobile Internet access
  - Mobile devices
- **Network Computing**
  - Software-as-a-service business models
  - Web services interoperability
  - Distributed computing (P2P, grid computing, hosted "cloud computing" server farms such as Amazon S3)
- **Open Technologies**
  - Open API's and protocols
  - Open data formats
  - Open-source software platforms
  - Open data (Creative Commons, Open Data License, etc.)

Web 3.0: Convergence of Trends

- **Open Identity**
  - Open identity (OpenID)
  - Open reputation
  - Portable identity and personal data (for example, the ability to port your user account and search history from one service to another)
- **The Intelligent Web**
  - Semantic Web technologies (RDF, OWL, SWRL, SPARQL, Semantic application platforms, and statement-based datastores such as triplestores, tuplestores and associative databases)
  - Distributed databases -- "The World Wide Database" (wide-area distributed database interoperability enabled by Semantic Web technologies)
  - Intelligent applications (natural language processing, machine learning, machine reasoning, autonomous agents)

Web 1.0 vs. Web 2.0 vs. Web 3.0

- Think of Web 1.0 as a library. You can use it as a source of information, but you can't contribute to or change the information in any way.
- Web 2.0 is more like a big group of friends and acquaintances, uses the Internet to make connections between people. It is the input/output web.
- Web 3.0 will use the Internet to make connections with information. It will make the web more intelligent
  - Plug in your tastes and habits!

Deusto tech
Tecnológico Fundación Deusto
Teknologikoa Deustu Fundazioa
Deusto Technology Foundation

45

Universidad de Deusto
Deustuko Unibertsitatea

Web 3.0 Features

- Web 3.0 is going to be like having a **personal assistant** who knows practically everything about you and can access all the information on the Internet to answer any question
- Web 3.0 will provide users with **richer and more relevant experiences**
- With Web 3.0, every user will have a **unique Internet profile based on that user's browsing history**.
- Web 3.0 will use this profile to tailor the browsing experience to each individual. That means that if two different people each performed an **Internet search** with the same keywords using the same service, they'd receive different results **determined by their individual profiles and context**
- The foundation for Web 3.0 will be **application programming interfaces** (**APIs**).
  - Creating mashups will be so easy in Web 3.0 that anyone will be able to do it.
  - Web 3.0 will let users combine widgets together to make mashups by just clicking and dragging a couple of icons into a box on a Web page.

Deusto tech
Tecnológico Fundación Deusto
Teknologikoa Deustu Fundazioa
Deusto Technology Foundation

46

Universidad de Deusto
Deustuko Unibertsitatea

Semantic Approach to Web 3.0

- Some experts think that Web 3.0 will start fresh. Instead of using HTML as the basic coding language, it will rely on some new language.
- With the Semantic Web, computers will scan and interpret information on Web pages using **software agents**.
  - These software agents will be programs that crawl through the Web, searching for relevant information.
  - They'll be able to do that because the Semantic Web will have collections of information called **ontologies**.
    - An ontology is a file that defines the relationships among a group of terms. For example, the term "cousin" refers to the familial relationship between two people who share one set of grandparents

Web Browser Evolution

- **Traditionally** Web browsers have run on **desktop computers** connected to the Internet via LANs or modems
  - **Mobiles** have **evolved** to support data services and now include small web browsers
- Markup (HTML) and scripting (JavaScript) doesn't have to be used only for accessing the World Wide Web
  - It can now be used for **user interfaces for many kinds** of **local and remote capabilitie**s, cutting development costs over traditional approaches to application development
  - "**Browsers**" are now a generic **user interface component**
- **Increasing variety of devices** being connected to networks, not just desktops and mobile phones
  - Web applications need to encompass a much wider range of devices
- A more powerful framework for Web applications would enable the removal of current Internet walls
  - Browser as a global interaction enabler

Future Browsers

- **What will be the role of browsers in the Future?**
  - How will future browsers interact with sites and applications?
  - Will we even have browsers or will everyone use browser-like single-purpose applications to access different areas of the Web?
- Mozilla, Opera and others are working toward a world in which the **browser is the operating system**—call it the browser as OS or the cloud OS.
  - All of your applications, documents, files and services are accessed from a Web interface and behave the same no matter what the underlying operating system (Windows, Linux, Android—whatever).

Future Browser Features

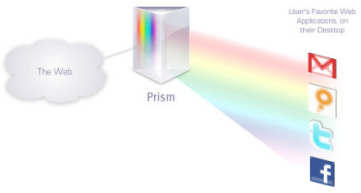
- They will be web application players:
  - Application cache
  - Offline/online operation
  - Localserver
  - Storage
  - Threading
  - Integration with desktop

The Browser as an Information Broker

- New role for Web Browsers:
  - **Detecting information in Web pages and handing that information off to other applications**
    - From HTML renderer to being an *information broker*.
- Microformat detection should be designed as a completely open and extensible platform.
  - Contact management, calendaring, and mapping applications use browser API to integrate with browser's microformat detection system.

Mozilla Labs Prism

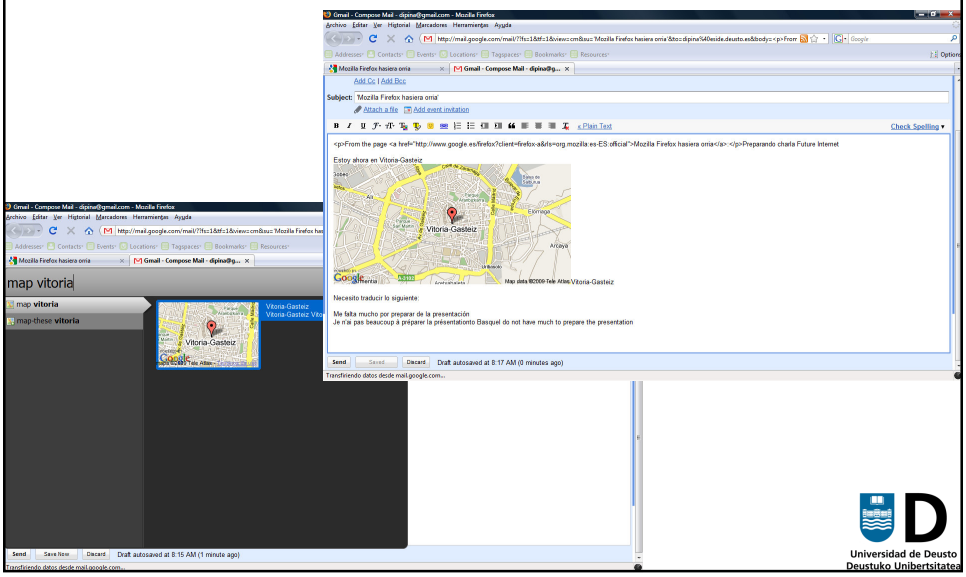
- Prism is an application that lets users split web applications out of their browser and run them directly on their desktop.
- Prism isn't a new platform, it's simply the web platform integrated into the desktop experience.
  - Prism lets users add their favorite web apps to their desktop environment
  - When invoked, these applications run in their own window:
  - Increase the capabilities of those apps by adding functionality to the Web itself, such as providing support for offline data storage and access to 3D graphics hardware.
    - http://labs.mozilla.com/2007/10/prism/



Mozilla Ubiquity

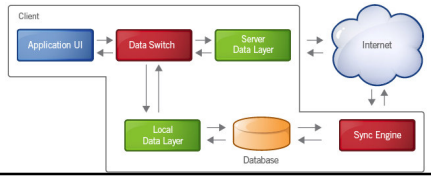
- **An experiment in connecting the web with language**
  - With Ubiquity installed, you'll be able to tell Firefox *what you want it to do* by typing **commands** into a new Ubiquity input box.
    - Ubiquity comes with a set of commands that make common Web tasks faster and easier.
  - Allows you to create user generated mash-ups
- Issue commands directly or previously selecting text pieces:
  - Ctrl+Space → displays command window
  - wiki deusto → searches for Deusto in Wikipedia
  - translate (text) (from language) (to language)
  - map [location | selected text]
  - command-list → enables you to request help from all the commands

Mozilla Ubiquity



Google Gears

- **Gears**, is a software which allows the development of more powerful applications, adding the following features to a browser:
  - A database module, based on SQLite, which allows storing data in the browser
  - A multithreading library (WorkerPool) which allows the execution in parallel of JavaScript code
  - A module named LocalServer which caches previously obtained resources from a web server (HTML, JavaScript, images, etc).
  - A desktop module which allows applications to interact with the local machine
  - A geolocation module which allows web applications to detect the geographical location of their users
    - http://gears.google.com/
    - http://code.google.com/intl/es-ES/apis/gears/sample.html

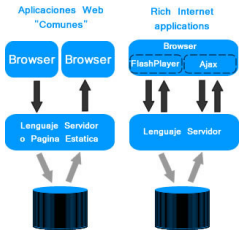


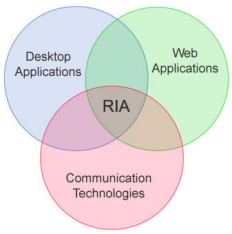
Deusto tech
Tecnológico Fundación Deusto
Teknologikoa Deustu Fundazioa
Deusto Technology Foundation

Universidad de Deusto
Deustuko Unibertsitatea

Rich Internet Applications (RIA)

- A current industry trend is improving the end-user experience while consuming SOA services
- This has given place to the so-called rich internet applications (RIA)
  - They combine the responsiveness and interactivity of desktop applications with the wide scope and ease of deployment of web applications





Deusto tech
Tecnológico Fundación Deusto
Teknologikoa Deustu Fundazioa
Deusto Technology Foundation

56

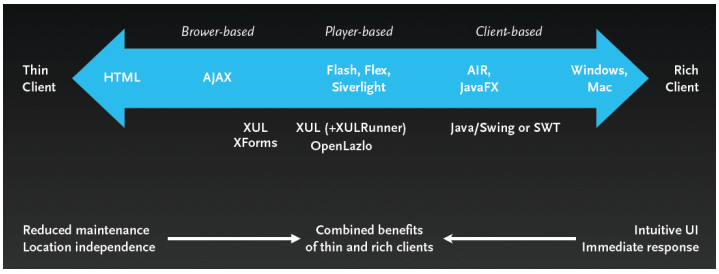
Universidad de Deusto
Deustuko Unibertsitatea

RIA Features

- They are web applications with a similar functionality to the traditional desktop-based applications
- They are client applications which maintain state and are decoupled from the service layer in the back-end
- RIA apps require the following execution environment:
  - They are executed in a web browser, they do not require instalation
  - They are executed in a secure environment denominated sandbox
- Advantages:
  - They allow users to find information more easily
  - They allow completing tasks more rapidly and in a precise manner
  - They generate rich data visualizations to help in the decision making process

Taxonomy of RIA Applications

- The *declarative model* of the new RIA tools — and Flex, Laszlo, XUL, and XAML-based platforms in particular — is an amazing advance that changes the model for developing software from a "*how*" model to a "*what*" model.
- 3 types:
  - **Client-based** – applications based on a client desktop
  - **Player-based** – applications based on an extension of the player
  - **Browser-based** – applications using JavaScript framework



New Generation of Web Apps

- The next-gen web is starting to gather pace thanks to technologies like AIR, Silverlight, JavaFX, Gears, XUL, Web Applications 1.0 (HTML 5.0)
  - Allow developers to accelerate beyond AJAX and **towards a new generation of web applications with better performance, more functionality and tighter desktop integration**
    - **"DLL hell" has been superseded by "plug-in hell"**
      - There is an opportunity/need to not repeat the mistakes of the past and instead take a standards-based approach
        - » http://www.techcrunch.com/2008/05/29/the-next-gen-web-browser-storage-support/

HTML 5.0

- Web Hypertext Application Technology Working Group (WHATWG) was formed in 2004 to extend HTML with other standards such as Xforms
  - HTML 5.0 specification (www.w3.org/html/wg/html5):
    - "The main area that has not been adequately addressed by HTML is Web Applications. This specification attempts to rectify this, while at the same time updating the HTML specs to address issues raised in the past few years"
- HTML 5 aims to become a game-changer in Web application development, one that might even make obsolete such plug-in-based rich Internet application (RIA) technologies
- HTML 5 provides a number of new elements and attributes that reflect typical usage on modern Web sites.
  - Some of them are semantic replacements for common uses of generic block ( ) and inline () elements, for example <nav> (website navigation block) and <footer>.
  - Other elements provide new functionality through a standardized interface, such as the <audio> and <video> elements
  - Some deprecated elements from HTML 4.01 have been dropped, including purely presentational elements such as <font> and , whose effects are achieved using CSS.
- There is also a renewed emphasis on the importance of DOM scripting in Web behavior.

HTML 5.0 Features

- HTML 5.0 introduces new document structure elements such as header, nav, article, section, aside, and footer.
  - Aside – offers support for sidebars
  - Datagrid
  - Dialog
  - Figure
  - Web Forms 2.0
- The real power of HTML 5 will be in its new APIs for authoring of Web-based applications :
  - WebSocket interface will enable to create bidirectional TCP/IP sockets between the application and the server
  - Remote-Events service allows servers to send document-objet-model (DOM) events to applications
  - Notitication service lets the server notify the user of new events, such as newly arrived email
  - New local storage API will make it possible for user agents to store and retrieve much more information that can be created as simple name-value pairs as in cookies.
- HTML 5.0 includes explicit support for offline execution of Web applications: application cache and application cache manifest (what the server needs to the user agent cache)
- New multimedia features such as 2D Graphics and audio/video control APIs
- Webworker and geolocation specifications

HTML 5.0 Deployment

- It will probably be years before we encounter support for HTML 5.0 in browsers
- In the mean time:
  - Opera and Webkit are actively involved implementing parts of HTML 5.0
  - Google Gears (http://code.google.com/apis/gears) adds new features reminiscent of HTML 5 to your Web browser:
    - APIs for geolocation
    - Client-side storages
    - Offline operation
    - Web workers
- There are clear similarities between Gears and HTML 5.0
  - Fortunately, Google engineers are active contributors of related W3C specifications.
- The combination of HTML and JavaScript is becoming an increasingly powerful application platform, in fact, it is becoming the “web application platform” for the front-end

Mobile RIA

- Adobe FlashLite
- Microsoft Silverlight with WPF for mobiles
- Mobile Ajax
- Gears: open Source, published by Google for Windows Mobile and Android
- Nokia Web Run Time: based on widgets to access info from Internet, bind it to user data kept in mobile and show only things related to their context

Mobile RIA

- iPhone's success and Android's interest have created new interest on application platforms for mobile phones
- Mobile web browsers are getting really good:
  - Nokia, Apple, Opera and Microsoft best mobile web browsers can handle any content
- Over the years, Web 2.0 technologies such as XMLHttpRequest or Falsh have created powerful web-based applications
  - Now it is time to take advantage of this apps on smart phones
- Webkit (http://webkit.org), Opera, Mozilla and Internet Explorer provide support for high-performance JavaScript execution.

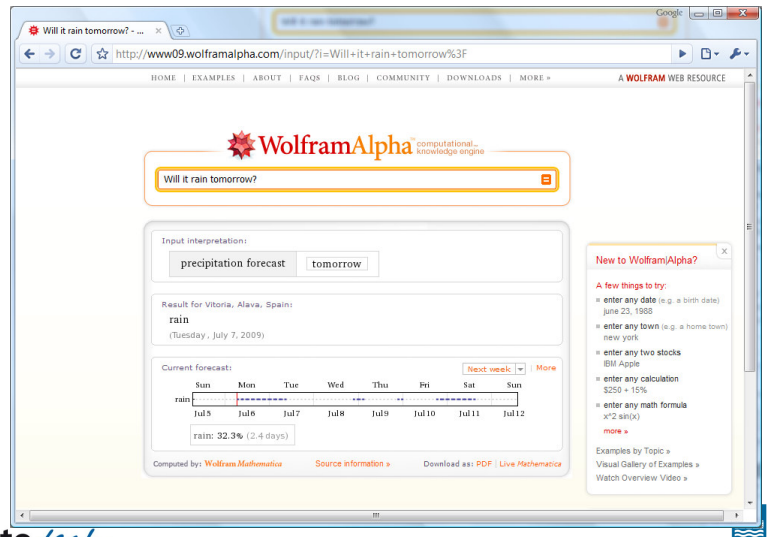
Future of Searching

- Lately, some very interesting search-engine-like services have emerged:
  - Wolfram|Alpha
  - Microsoft Bing and
  - Google Squared

Wolfram|Alpha

- It's a **computational knowledge engine**: it generates output by doing computations from its own internal knowledge base, instead of searching the web and returning links.
  - Attempt to gather facts, formulas, and natural language processing algorithms, encode them all in the language of Mathematica
  - The results come back as graphs, maps, simple facts, and tables
- **So what is Wolfram|Alpha good for?**
  - Fact and computations about facts.
  - The service already knows a lot about the world:
    - Geography, economics and socioeconomic data, physics, chemistry, engineering, sports, units of measurement, weather, and even music.
- Differences with Search Engines:
  - All of the facts, formulas and relationships between data points are curated by real humans
  - Wolfram|Alpha **tries to determine the real meaning of the question being asked**.
    - Natural language parsers attempt to determine the domain(s) of the question, and that becomes part of the computation for the answer.

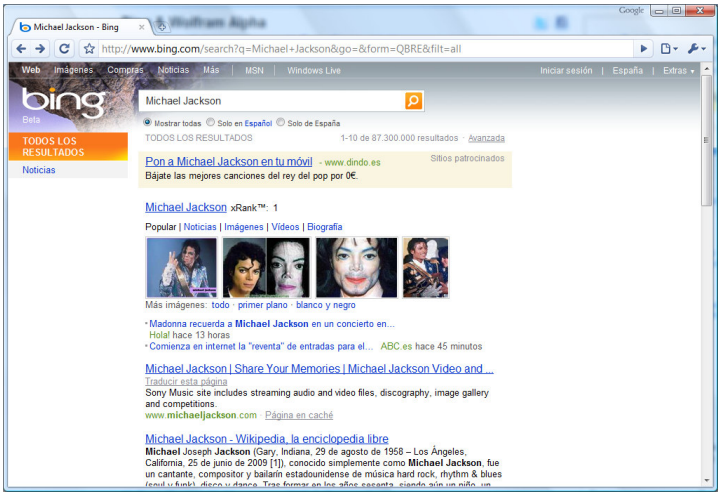
Wolfram|Alpha



Microsoft Bing

- Bing is a replacement for Live Search.
- Microsoft's new search engine, makes an extra effort to help searchers with some targeted topics: shopping, travel, local business and information, and health-related research as areas in which people wanted more assistance in making key decisions
- Some tweaks over the competition:
  - Listing of search suggestions in real-time
  - List of search refinements along the left side
  - The left-side bar also shows suggested "related searches"
  - The interface also has other usability improvements — thumbnails of videos will start playing when you mouse over them, the image search results employ "infinite scrolling"

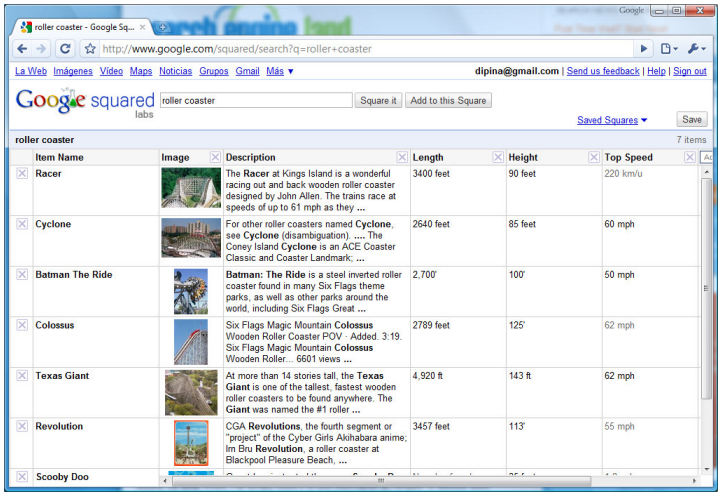
Microsoft Bing



Google Squared

- "Squared" refers to its ability construct a table of facts from two search terms, similar in result to a spreadsheet.
  - On one axis you can put a general search term — say "roller coasters" and across the other axis add headings that describe the facts you want to know about the search term — such as height and speed.
  - The result is a two dimensional table of results.
- In contrasted with Wolfram|Alpha, though — where all of the information is specifically curated to link up to each other — the facts in Google Squared are collected from the web

Using Google Squared



Web as a Platform: Client-side

- Technologies such as AIR, XUL, BrowserPlus or Gears share the worthy aim of allowing "a new generation of web applications with better performance, more functionality and tighter desktop integration
  - XUL, Flex, Silverlight allow you to create advanced RIA apps
  - Air essentially brings the browser offline
  - BrowserPlus runs outside of the browser to make your desktop an extension of the web
  - Gears runs inside the browser, making Firefox even more unstable, but does make my web browsing faster
- Some of the code behind these efforts will be opened up to the community and turn standard
  - In the meantime, "graceful degradation"
    - http://www.whatwg.org/specs/web-apps/2005-09-01/

Web as a Platform: Server-side

SOA: Service Oriented Arquitecture

- Define la utilización de **servicios** para dar soporte a los **requisitos del negocio**
- Sistemas altamente **escalables**, estándar de exposición e invocación de servicios (comúnmente pero no exclusivamente servicios web), lo cual **facilita la interacción entre diferentes sistemas** propios o de terceros.
- **Capas:**
  - Aplicaciones básicas
  - De exposición de funcionalidades
  - De integración de servicios
  - De composición de procesos
  - De entrega
- Éxito de SOA depende de que los desarrolladores de software se orienten a esta mentalidad de crear **servicios comunes** que son orquestados por clientes o middleware para implementar los procesos de negocio
- Hablar de SOA significa generalmente hablar de un juego de servicios residentes en Internet o una intranet, usando servicios web (*XML*, *HTTP*, *SOAP*, *WSDL*, *UDDI*)

WOA: Web Oriented Arquitecture

- Arquitectura de software que extiende SOA a aplicaciones basadas en web
- La información se representa en forma de **recursos** en la red y es accedida y manipulada mediante el protocolo especificado en la URI.
- Los recursos se manipulan mediante variables HTTP (**GET**, **PUT**, **POST**, **DELETE**) usando la técnica **REST**.
- Estos recursos solo son manipulados por componentes pertenecientes a la red (esencialmente browsers & web servers).
- El acceso a los recursos no debe requerir más que un conocimiento local de la red.
- Es responsabilidad de los componentes entender la representación y estados de transición válidos de los recursos.
- Los recursos tienen embebidas URIs que construyen una red más grande de recursos (i.e. recursos de pedidos contienen URIs a recuros de inventario)

Web as a Platform: Server-side

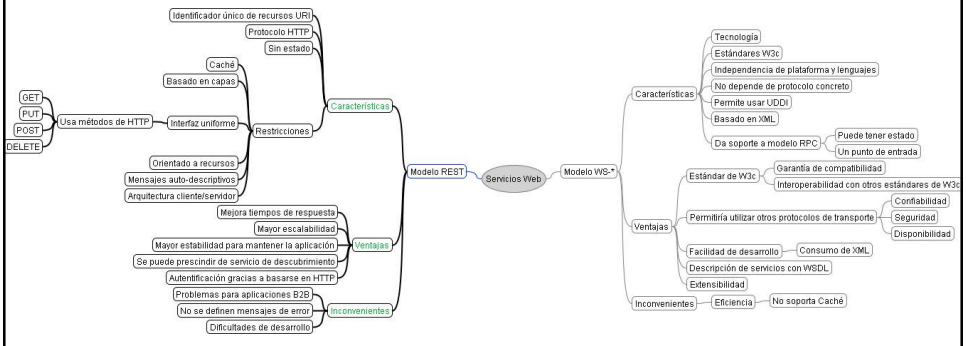
SOAP: Simple Object Access Protocol

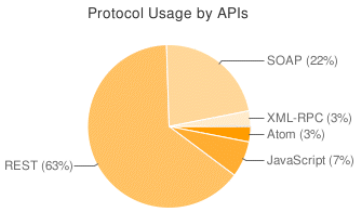
- Define como dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos **XML**.
- Deriva de un protocolo creado por David Winer en 1998, llamado **XML-RPC**.
- Fue creado por **Microsoft**, **IBM**, y otros y actualmente se encuentra bajo el auspicio de la **W3C**.
- Su arquitectura consiste en varias capas de especificaciones para formato de mensajes:
  - Message Exchange Patterns (MEP)
  - Protocolos de transporte (SMTP y HTTP/S)
  - Modelos de procesado de mensajes
  - Protocolo de extensibilidad

REST: Represantional State Transfer

- **Estilo** de arquitectura software para sistema de hypermedia distribuidos como la **WWW**.
- Introducido en la Tésis doctoral de **Roy Fielding** en el año 2000.
- Se refiere a una colección de principios de arquitectura de red, que marcan como definir e invocar los recursos.
- El término se usa a veces para describir una simple interfaz que transmite datos de un dominio específico por **HTTP** sin capas adicionales como SOAP o uso de cookies.
- Estas dos características pueden chocar o solaparse.
- Los sistemas que cumplen los principios marcados por Fielding suelen ser referidos como sistemas **RESTful**.

SOAP vs. REST





Mobile Servers

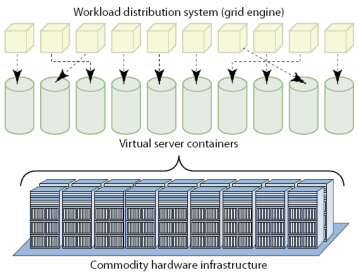
- A current trend is to have mobile applications serving applications:
  - Nokia Mobile Web Server
    - Ports apache httpd and mod_python to Nokia S60
    - Gateway to prevent problems from firewalls
    - http://mymobilesite.net/

Cloud Computing is …

- … virtualized compute power and storage delivered via platform-agnostic infrastructures of abstracted hardware and software accessed over the Internet. These shared, on-demand IT resources, are created and disposed of efficiently, are dynamically scalable through a variety of programmatic interfaces and are billed variably based on measurable usage.

Forrester Research

"A pool of abstracted, highly scalable, and managed compute infrastructure capable of hosting end-customer applications and billed by consumption[1]"



1- "Is Cloud Computing Ready for The Enterprise?" Forrester Research, Inc.

Forrester Research (cont'd)

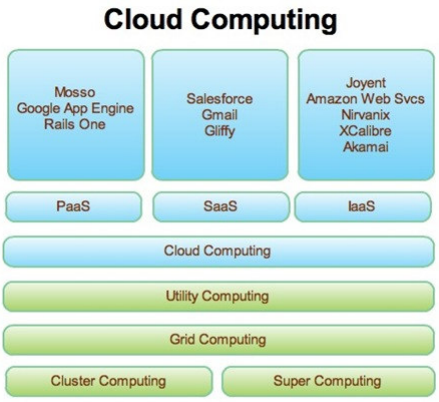
- Different than SaaS (Software as a Service)
  - Prescripted & Abstracted Infrastructure
  - Fully Virtualized
  - Dynamic Infrastructure Software
  - Pay by Consumption
  - Free of Long-Term Contracts
  - Application and OS Independent
  - Free of Software or Hardware Installation
- "Cloud computing has all the earmarks of being a potential disruptive innovation that all infrastructure and operations professionals should heed."

The "Cloud" = 10X Improvement

- **Ease of Use**: Do it yourself remotely from anywhere anytime
- **Scalability**: control your infrastructure with your app
- **Risk**: nothing to buy, cancel immediately
- **Reliability**: Based on enterprise grade hardware
- **Cost**: pay for only what you use

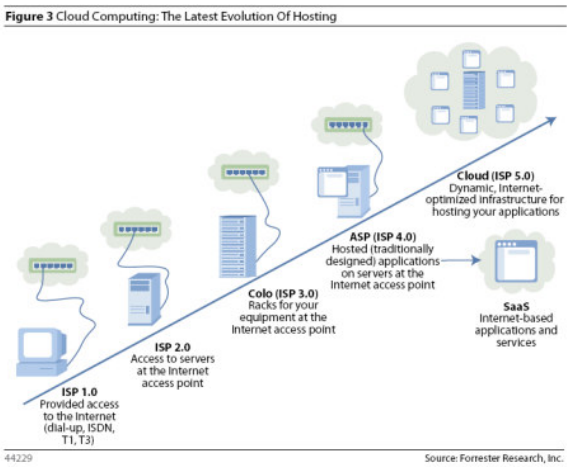
Evolution Towards Cloud Computing

- Cluster computing and supercomputing coexistence and limitations gave place to grid computing
- From grid computing we progressed towards utility computing, i.e. Packetize computation services such as light, water, and so on.
- This derived into Cloud Computing, divided into:
  - *Plataform as Service*
  - *Software as Service*
  - *Infraestructure as Service*

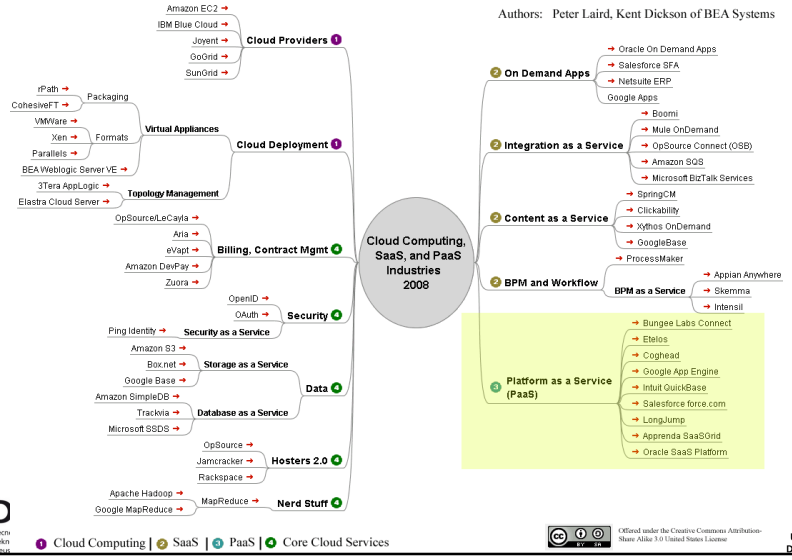


1.0 In blue you have what is lately called Cloud Computing. In green, some of the underlying work done that led to Cloud Computing. At the top are examples of each XaaS type.

Multiple Graphic Descriptions of the "Cloud"



Multiple Graphic Descriptions of the "Cloud"



Cloud Features

Deployment Types

- **Private cloud**
  - Enterprise owned or leased e.g., in case of data centers, HPC centers,…
- **Community cloud**
  - Shared infrastructure for specific community
- **Public cloud**
  - Sold to the public, mega-scale infrastructure e.g., ec2, S3,…
- **Hybrid cloud**
  - Composition of two or more clouds

Delivery Models

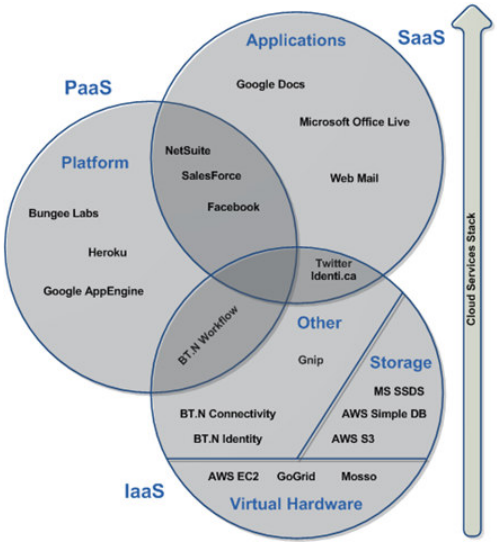
- **Cloud Software as a Service (SaaS)**
  - Use provider's applications over a network e.g., Salesforce.com,…
- **Cloud Platform as a Service (PaaS)**
  - Deploy customer-created applications to a cloud e.g., Google App Engine, Microsoft Azure, …
- **Cloud Infrastructure as a Service (IaaS)**
  - Rent processing, storage, network capacity, and other fundamental computing resources e.g., EC2 – Elastic Computer Cloud, S3 – Simple Storage Service, Simple DB,…

Deusto tech
Tecnológico Fundación Deusto
Teknologikoa Deustu Fundazioa
Deusto Technology Foundation

84

Universidad de Deusto
Deustuko Unibertsitatea

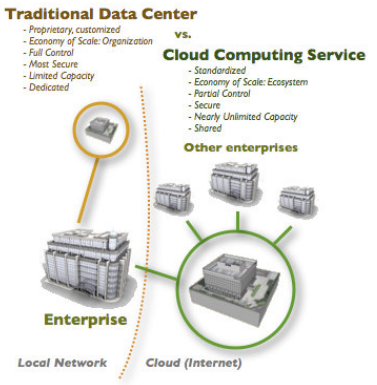
Delivery Models

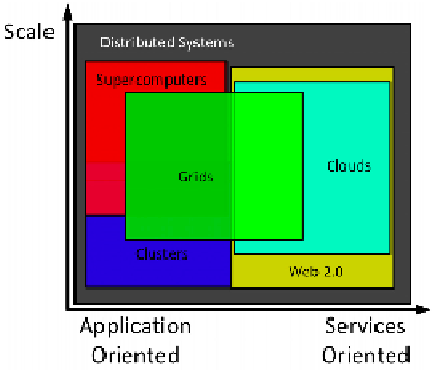


Deusto tech
Tecnológico Fundación Deusto
Teknologikoa Deustu Fundazioa
Deusto Technology Foundation

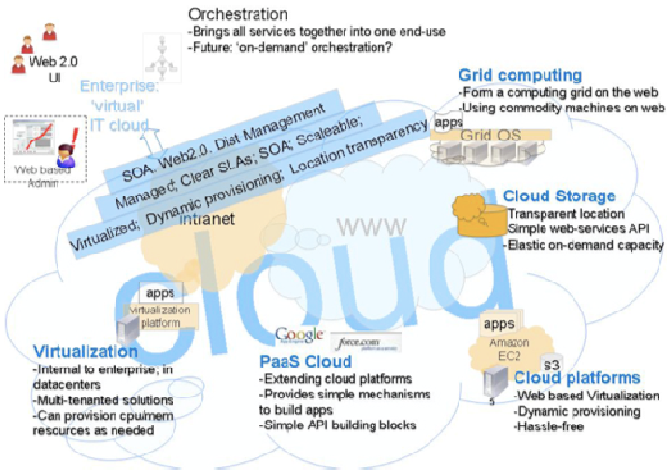
Universidad de Deusto
Deustuko Unibertsitatea

Cloud Computing vs. Traditional Data Centres





Cloud Computing Components



The Cloud's "Snowball Effect"

- Maturation of Virtualization Technology
- Virtualization enables Compute Clouds
- Compute Clouds create demand for Storage Clouds
- Storage + Compute Clouds create Cloud Infrastructure
- Cloud Infrastructure enables Cloud Platforms & Applications
- Multiple Cloud types lead to Cloud Aggregators
- Niche requirements enable Cloud Extenders

SaaS: Software as a Service

- Software delivery model:
  - Increasingly popular with SMEs
  - No hardware or software to manage
  - Service delivered through a browser
- Advantages:
  - Pay per use
  - Instant Scalability
  - Security
  - Reliability
  - APIs
- Examples: CRM, Financial Planning, Human Resources, Word processing
- Commercial Services: Salesforce.com, emailcloud

PaaS: Platform as a Service

- Platform delivery model:
  - Platforms are built upon Infrastructure, which is expensive
  - Estimating demand is not a science!
  - Platform management is not fun!
- Advantages:
  - Pay per use
  - Instant Scalability
  - Security
  - Reliability
  - APIs
- Examples: Google App Engine, Mosso, AWS: S3
- Popular Services: Storage, Database, Scalability

IaaS: Infrastructure as a Service

- Computer infrastructure delivery model:
  - Access to infrastructure stack:
    - Full OS access
    - Firewalls
    - Routers
    - Load balancing
- Advantages:
  - Pay per use
  - Instant Scalability
  - Security
  - Reliability
  - APIs
- Examples: Flexiscale, AWS: EC2

Cloud "Applications"

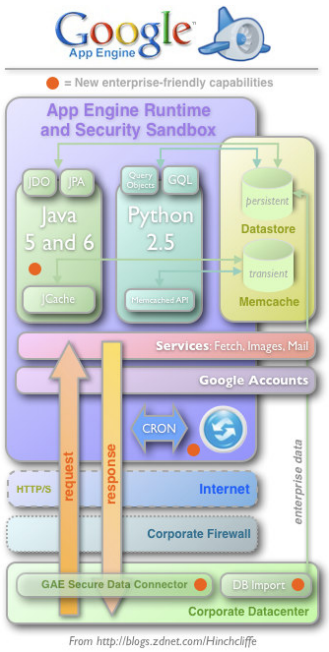
- SaaS resides here
- Most common Cloud / Many providers of different services
- Examples: SalesForce, Gmail, Yahoo! Mail, Quicken Online
- Advantages: Free, Easy, Consumer Adoption
- Disadvantages: Limited functionality, no control or access to underlying technology

Cloud "Platforms"

- "Containers"
- "Closed" environments
- Examples: Google App Engine, Heroku, Mosso, Engine Yard, Joyent or Force.com (SalesForce Dev Platform)
- Advantages: Good for developers, more control than "Application" Clouds, tightly configured
- Disadvantages: Restricted to what is available, other dependencies

heroku MOSSO the hosting cloud Google App Engine

force.com platform as a service Joyent Engine Yard



From http://blogs.zdnet.com/Hinchcliffe

Cloud "Infrastructure"

- Provide "Compute" and "Storage" clouds
- Virtualization layers (hardware/software)
- Examples: Amazon EC2, GoGrid, Amazon S3, Nirvanix, Linode
- Advantages: Full control of environments and infrastructure
- Disadvantages: premium price point, limited competition

Cloud "Extenders" (Wild Card)

- Provides extension to Cloud Infrastructure and Platforms with basic functionality
- Examples: Amazon SimpleDB, Amazon SQS, Google BigTable
- Advantages: Extends functionality of Compute & Storage Clouds to integrate with legacy system or other clouds
- Disadvantages: Sometimes requires use of specific Platforms or Infrastructure

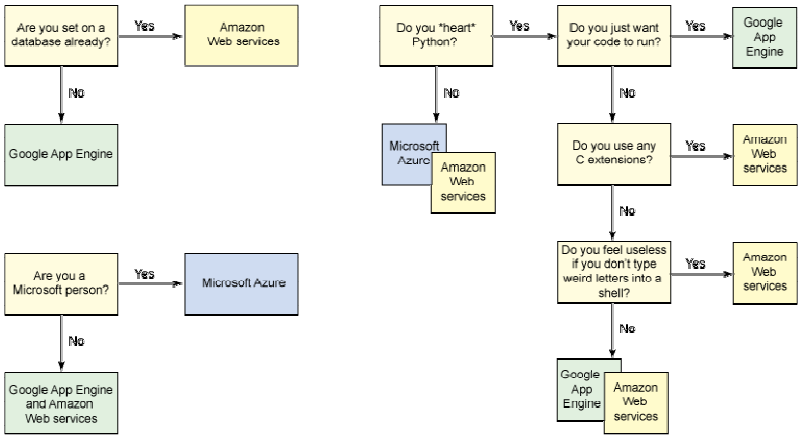
Cloud "Aggregators" (Wild Card)

- Sits on top of various Cloud Infrastructures for management
- Examples: RightScale, Appistry
- Advantages: Provides more options for Cloud environments
- Disadvantages: Dependent on Cloud Providers

Existing Platforms

	Amazon Elastic Compute Cloud (EC2)	Google App Engine	Microsoft Azure	Sun Network.com (Sun Grid)	Grid Lab Aneka
Focus	Infrastructure	Platform	Plataform	Infrastructure	Software Platform for enterprise Clouds
Service Type	Computer, Storage (Amazon S3)	Web Application	Computer, Storage	Compute	Compute
Virtualisation	OS Level running on a Xen hypervisor	Application container	Windows Azure OS	Job management system (Sun Grid Engine)	Resource Manager and Scheduler
Dynamic Negotation of QoS Parametres	None	None	None	None	SLA-based Resource Reservation on Aneka side
User Acces Interface	Amazon EC2 Command-line Tools	Web-based Administration Console	Web-based Live Desktop and any devices	Job submission scripts, Sun Grid Web portal	Workbench, Web-based portal
Web APIs	Yes	Yes	Yes	Yes	Yes
Value-added Service Providers	Yes	No	No	Yes	No
Programming Framework	Customizable Linux-based Amazon Machine Image (AMI)	Python	.Net Framework	Solaris OS, Java, C, C++, FORTRAN	APIs supporting different programming models in C# and other .Net lengauges

Which one to use?

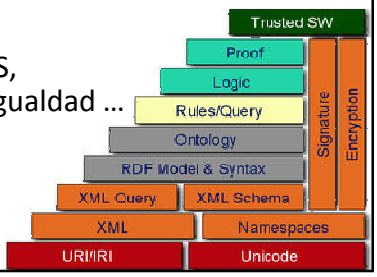


Web Semántica

- Problema de la Web Actual:
  - El significado de la web no es accesible a máquinas
- **Web Semántica** → crea un medio universal de intercambio de información, aportando semántica a los documentos en la web
  - Añade significado comprensible por ordenadores a la Web
  - Usa técnicas inteligentes que explotan esa semántica
  - Liderada por Tim Berners-Lee del W3C
- **Mission: turning existing web content into machine-readable content**

Semantic Web Stack

- La Web Semántica está compuesta de:
  - XML, sintaxis para documentos estructurados
  - XML Schema, restringe la estructura de documentos XML
  - RDF es un modelo de datos que hace referencia a objetos y sus relaciones
  - RDF Schema, vocabulario para definir propiedades y clases de recursos RDF
  - OWL, añade más vocabulario que RDFS, relaciones entre clases, cardinalidad, igualdad …



RDF Resource, Property, and Property Value

- RDF identifies things using Web identifiers (URIs), and describes resources with properties and property values.
- Explanation of Resource, Property, and Property value:
  - A **Resource** is anything that can have a URI, such as "http://www.w3schools.com/RDF"
  - A **Property** is a Resource that has a name, such as "author" or "homepage"
  - A **Property value** is the value of a Property, such as "Jan Egil Refsnes" or "http://www.w3schools.com" (note that a property value can be another resource)

Resource Description Framework (RDF)

- Un grafo RDF crea una web de conceptos
  - Realiza aserciones sobre relaciones lógicas entre entidades
- Información en RDF puede ligarse con grafos en otros lugares
  - Mediante software se pueden realizar inferencias
    - Lenguajes de consulta sobre triple stores como SPARQL
- Mediante RDF hacemos que la información sea procesable por máquinas
  - Agentes software pueden guardar, intercambiar y utilizar metadatos sobre recursos en la web
- **Ontología** → jerarquía de términos a utilizar en etiquetado de recursos

Example RDF

- **Formato RDF/XML:**

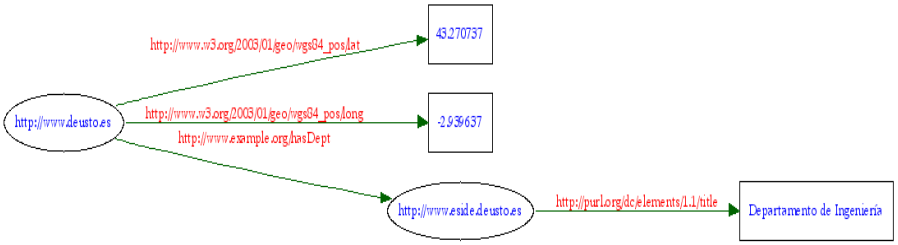
```
1: <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
2:    xmlns:dc="http://purl.org/dc/elements/1.1/"
3:    xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos/"
4:    xmlns:edu="http://www.example.org/">
5:    <rdf:Description rdf:about="http://www.deusto.es">
6:       <geo:lat>43.270737</geo:lat>
7:       <geo:long>-2.939637</geo:long>
8:       <edu:hasFaculty>
9:          <rdf:Bag>
10:            <rdf:li rdf:resource="http://www.eside.deusto.es" dc:title="Facultad de
    Ingeniería"/>
11:            <rdf:li rdf:resource="http://www.lacomercial.deusto.es" dc:title="Facultad de
    Económicas y Empresariales"/>
12:            </rdf:Bag>
13:      </edu:hasFaculty>
14:   </rdf:Description>
15: </rdf:RDF>
```

- **N3/Turtle:**

```
1: @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
2: @prefix dc: <http://purl.org/dc/elements/1.1/> .
3: @prefix geo: <http://www. w3.org/2003/01/geo/wgs84_pos#> .
4: @prefix edu: <http://www.example.org/> .
5: <http://www.deusto.es> geo:lat "43.270737" ; geo:long "-2.939637" .
6: <http://www.eside.deusto.es> dc:title "Departamento de Ingeniería" .
7: <http://www.deusto.es> edu:hasFaculty <http://www.eside.deusto.es> .
```

RDF Example

- http://www.w3.org/RDF/Validator/



What is an ontology?

- An ontology describes basic concepts in a domain and defines relations among them. Basic building blocks of ontology design include:
  - classes or concepts
  - properties of each concept describing various features and attributes of the concept (slots (sometimes called roles or properties))
  - restrictions on slots (facets (sometimes called role restrictions))
- An ontology together with a set of individual instances of classes constitutes a knowledge base.

OWL Review

- An ontology differs from an XML schema in that it is a knowledge representation, not a message format
- One advantage of OWL ontologies will be the availability of tools that can reason about them
- The normative OWL exchange syntax is RDF/XML.
- OWL is a vocabulary extension of RDF
- Web ontologies are distributed
- Can be imported and augmented, creating derived ontologies.

Ejemplo OWL Ontology Reasoning

- Supongamos el siguiente modelo RDF:

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
<http://www.ipina.org/> foaf:author
   <http://www.ipina.org/osgi/> .
<http://www.deusto.es/dipina/> foaf:author
   <http://www.deusto.es/dipina/ajax/> .
<http://www.eside.deusto.es/dipina/> foaf:author
   <http://paginaspesonales.deusto.es/dipina/> .
```

- Aunque pertenecen al mismo autor, no están relacionadas entre ellas, con la ayuda de OWL podemos mapear estas URIs

```
@prefix owl: <http://www.w3.org/2002/07/owl#> .
<http://www.deusto.es/dipina/> owl:sameAs
   <http://www.ipina.org/> .
<http://www.eside.deusto.es/dipina/> owl:sameAs
   <http://www.ipina.org/> .
```

- Si mezclamos ambos modelos y ejecutamos un razonador podríamos responder a "dime todo lo que ha escrito "<http://www.ipina.org>":

```
<http://www.ipina.org/osgi/>,<http://www.deusto.es/dipina/aja
   x/> y <http://paginaspesonales.deusto.es/dipina/>
```

SPARQL

- SPARQL (http://www.w3.org/TR/rdf-sparql-query/) permite la consulta de grafos RDF a través de un lenguaje sencillo
- SPARQL es idóneo para extraer y consultar información mantenida por aplicaciones, servicios o repositorios ad-hoc de terceras partes expresados en RDF
- Consta de 3 elementos:
  - Lenguaje de consultas.
  - Mecanismo para transmitir una consulta a un servicio de procesamiento de consultas remoto
  - Formato XML en el que devolver los resultados

SPARQL Example

```
PREFIX table: <http://www.daml.org/2003/01/periodictable/PeriodicTable#>
SELECT ?symbol ?number
FROM <http://www.daml.org/2003/01/periodictable/PeriodicTable#>
WHERE
{
  {
  ?element table:symbol ?symbol;
           table:atomicNumber ?number;
           table:group table:group_17.
  OPTIONAL { ?element table:color ?color. }
  }
  UNION
  {
  ?element table:symbol ?symbol;
           table:atomicNumber ?number;
           table:group table:group_18.
  }
}
ORDER BY DESC(?number)
LIMIT 10
OFFSET 10
```

RDFa

- Problems with RDF:
  - RDF is a flexible format for storing, aggregating, and using metadata.
  - BUT, RDF/XML syntax is messy enough to scare many people away from RDF
    - SOLUTION → RDFa
- **RDFa = Making easier to embed RDF into XHTML and XML**
  - Adds metadata to XHTML without affecting browsers display
  - Web page data is readable not only by humans but by automated processes
  - Enables data aggregation and metadata association to perform tasks more sophisticated than those enabled by screen scrapping
  - Uses some existing XHTML 1 attributes and a few new XHTML 2 attributes to store the subjects, predicates, and objects of RDF triples
    - XHTML 1 attributes `href`, `content`, `rel`, `rev`, and `datatype`
    - New `about`, `role` and `property` attributes from XHTML 2's Metainformation Attributes (http://www.w3.org/TR/xhtml2/mod-meta.html) module
- Good site: http://rdfa.info/

GRDDL

- **GRDDL** is a mechanism for Gleaning Resource Descriptions from Dialects of **L**anguages
  - Defines a standard for declaring that a web page or XML can be transformed into an RDF graph, as well as the algorithms o mechanisms for performing such transformations
    - `glean –verb (used with object)`
      `1. to learn, discover, or find out, usually little by little or slowly.`
      `–verb (used without object)`
    - Joins the gap between microformats and RDFa
- A markup for
  - ... declaring that an XML document includes gleanable data (PROFILE) or
  - ... linking to an algorithm (typically XSLT) for gleaning the RDF data from the document (TRANSFORMATION)
- The markup includes:
  1. a namespace-qualified attribute for use in general-purpose XML documents.
     ```
     xmlns:grddl='http://www.w3.org/2003/g/data-view#'
     grddl:transformation="glean_title.xsl"
     ```
  2. a profile-qualified link relationship for use in valid XHTML documents.
     ```
     <head profile="http://www.w3.org/2003/g/data-view">
     <link rel="transformation" href="glean_title.xsl" />
     ```

Summary Mechanisms to Add Semantics

- **Microformats**
  - Designed for humans first, machines second
  - Very short steps to solve specific problems one at a time
  - Help putting your data into HTML, but no standard way to get the data out.
  - Cannot be validated easily, mixing `hCard` and `hCal` there's no way to guarantee you will interpret it correctly.
    - Very domain specific
    - `<a href="http://jane-blog.example.org/" rel="sweetheart date met">Jane</a>`
- **RDFa vs. microformats**
  - when the underlying ontology/vocabulary is simply way too complicated to be re-expressed in a microformat
    - microformats are good for micro-metadata (e.g. name, address, event dates)
    - but not for complex embedded (e.g. proteins, geological data)
  - when you need to combine several ontologies/vocabularies in one page
    - handle possible conflicts between primitives e.g. class="name" and class="name" vs. rel="foaf:name" and rel="prod:name"
- **GRDDL**
  - Middleware to join microformats and RDFa
    - Glue for the different existing semantization mechanisms
- **All of them together will help us rich the vision of a more meaningful, still painless Web!!**

Semantic Web vs. semantic web

	Semantic Web	semantic web
Philosophy	Build a common data format for expressing the meaning of data. Use ontologies to help machines to understand web content.	Humans first, machines second. Encode existing Web content with special tags.
Language	**RDF, RDFS, OWL**	**Microformats (based on XHTML), RDFa, GRDDL**
Format	Must be well-formed RDF documents	Anything goes, as long as its XHTML
Semantic	Defined by the underlying ontology model (e.g., OWL)	Loosely defined. No formal semantic model, unless RDFa is used
Examples	FOAF, OWL-S, OWL-Time	XFN (social network), hCard (contact), hReview (opinions), rel-tag (tagging)

Web 2.0 and Mash-up Problems

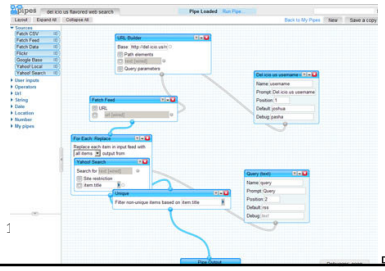
- Vanilla XML Web services require parsing XML trees to retrieve the desired data
  - Processing remains tied to underlying syntactic rather than semantic structure of the data
  - Need of writing custom handlers to interact with each API
  - No common language is available for querying and integrating such data sources
- **The challenge is to create truly flexible Web scale mash-ups**
  - SPARQL query language for the Semantic Web enables standardised access to distributed data sources.

Semantic Web Problems

- Faces similar challenges to Web 2.0 mash-ups if it is to reach widespread adoption
  - Few mechanisms exist that allow non-specialist users to contribute to the Semantic Web
  - Completing forms in a web browser content that is immediately usable on the Semantic Web
    - Without any user knowledge of RDF, ontologies or Semantic Web
  - Promote user semantic annotation with guided user input through the use of forms
    - Significant effort must be given to developing compelling interfaces able to display structured, linked data from across the Web
  - First attempts → dbpedia RDF-ising Wikipedia

Current ways to make a Mashup

- Hand Code
- Mash-up makers (http://mashable.com/2007/07/08/mashups/):
  - Google Mashup Editor
  - Yahoo! Pipes (http://pipes.yahoo.com)
  - IBM's QEDWiki
  - Microsoft PopFly (http://www.popfly.com)

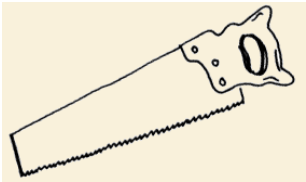


Problems of Current Tools

- A REST service can do anything
  - Any Input
  - Any Output
  - Multiple Ways to invoke
- REST can be invoked in different ways
- Data mediation must be done

SAWSDL

- **S**emantically **A**nnotated **WSDL**
- A W3C Recomendation
- A way of adding semantics to a WSDL
- Derived from METEOR-S (LSDIS project)
- Note that WSDL is in the name

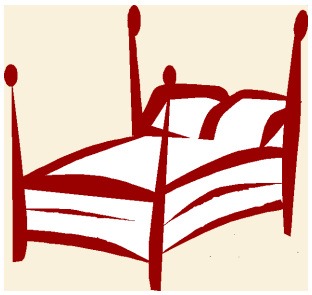


SAWSDL Details

- Add annotations through a "Model reference"
- Types of annotations
  - portTypes / Interfaces
  - Inputs or Outputs / Types
  - Faults / Faults
  - Operations
- Also have "Lifting Schemas"

SA-REST

- A way to attach Semantics to REST based services
- Based of SAWSDL
- Attempt to extend REST while keeping simplicity in mind



What could semantics fix

- Work with high level concepts
- Creates the possibility for homogeneity
- Passing of concepts not low level data structure
- Adds possibility for advanced search

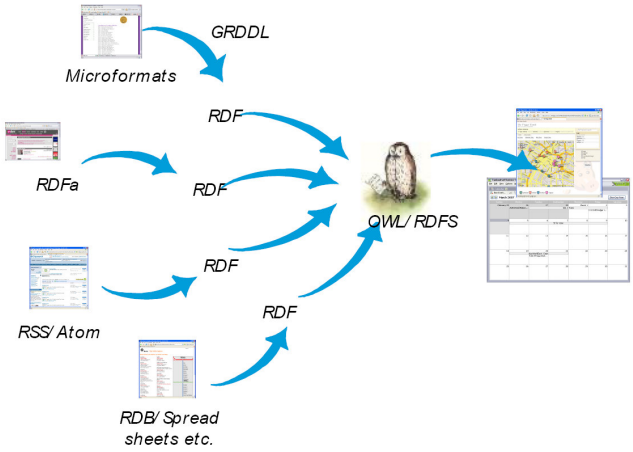
SA-REST Example

```
<html xmlns:sarest="http://lsdis.cs.uga.edu/SAREST#">
   <body>
      <p about="http://lsdis.cs.uga.edu/~jlathem/FindPerson">
         This service takes a
         <span property="sarest:input">
          http://xmlns.com/foaf/0.1/#person
         </span> as an input.
         The service returns a
         <span property="sarest:output">
          http://otheront/#location
         </span>
         as the result.
         The page should be invoked by an
         <meta property="sarest:catapouamus" content="HTMLGET"/>
         <meta property="sarest:upcast" content="test"/>
         <meta property="sarest:downcast" content="test2"/>
         <meta property="sarest:operation" content="LookupPerson"/>
      </p>
   </body>
</html>
```


Semantic Mash-ups

- SO FAR …
  - Mashups are great if you're looking for one kind of thing (coffee shops, hotels, gyms) and come from one source (especially when that source is an amalgamator like Citysearch or even Google)
- Semantic Mashups will merge information from multiple feeds and organize the results
  - First step is to turn all of your data sources into feeds that can be mashed up
    - RDFa lets Web site developers make an HTML page do double duty as a presentation page and as a machine-readable source of structured data in RDF.

Semantic Mash-ups



Mash-up Semánticos

- … o como la Web 2.0 actual puede beneficiarse de la adopción de ciertas tecnologías semánticas para dar lugar a portales web más inteligentes
  - Potencial combinación de Web 2.0 y Web Semántica
- Ejemplo:
  - mash-up semántico capaz de agregar información heterogénea sobre eventos proveniente de diferentes organizaciones y de mostrarla de manera conjunta sobre un mapa de GoogleMaps

Problemática Web 2.0 y Mashups

- La Web 2.0 social define un portal como un "jardín cerrado" (*walled garden*) donde sus frutos difícilmente pueden ser recogidos y combinados con los de otros portales.
- Una manera de romper los muros de los jardines de datos de los portales Web 2.0 actuales es ofrecer APIs
  - A menudo, estas APIs dan lugar a estructuras de datos en XML que deben ser procesadas e integradas por los consumidores
  - RDF permite combinar las respuestas de un portal con las de otro
    - haciendo que las sentencias RDF de dos localizaciones diferentes hagan referencia al mismo concepto (URI)
    - estableciendo correspondencias mediante OWL indicando que dos conceptos son equivalentes
- El uso de RDF para generar respuestas desde diferentes portales de datos distribuidos permite el uso del lenguaje de consultas SPARQL

Problemática de la Web Semántica

- La Web Semántica presenta dos problemas principales para su adopción global:
  - disponibilidad de los datos y
    - Necesidad de creación de interfaces que permitan a usuarios web convencionales contribuir a la Web Semántica
  - diseño de interfaces de interacción
    - Desarrollar interfaces que permitan visualizar más cómodamente los grafos de relaciones entre conceptos que constituyen la Web Semántica

Mash-ups normales vs. Semánticos

- En los mash-ups Web 2.0 tradicionales, **cada vez que se integra una nueva fuente de información es preciso desarrollar un nuevo adaptado**r que convierta los datos capturados al formato interno utilizado en el mash-up.
  - En el mash-up semántico basta rellenar un formulario web donde se especifica la URL de publicación de eventos de un portal, así como un conjunto mínimo de metadatos para automáticamente ser capaz de combinar los datos de este nuevo portal con los datos de otros portales antes considerados.
- Los **mash-up semánticos son mucho más flexibles** dado que convierten la información recuperada a formato RDF, *lingua franca* que puede luego ser fácilmente luego filtrada y consultada a través de consultas SPARQL.
- Los **mash-ups semánticos tienen la capacidad de evolucionar** sin requerir cambios en su código
  - Aunque los datos provistos pueden ser especificados en formatos de representación sintácticos diferentes, es un requisito fundamental que tales **datos, semánticamente, deben proveer una información muy similar**, fácilmente convertible a un vocabulario RDF común.

Ontología del Mash-up

- Ontología que modela "Eventos geo-localizados organizados por una entidad"
  - en vez de diseñar una ontología desde cero, hemos preferido hacer uso de tres vocabularios RDF bien conocidos tales como Calendar, vCard y Geo, que nos permiten representar eventos con las características que deseamos
    - RDF Calendar (http://www.w3.org/TR/2005/NOTE-rdfcal-20050929/)
    - vCard-RDF (http://www.w3.org/TR/vcard-rdf)
    - Geo RDF (http://www.w3.org/2003/01/geo/)
  - Y de sus microformatos equivalentes:
    - hCalendar (http://microformats.org/wiki/hcalendar)
    - hCard (http://microformats.org/wiki/hcard)
    - geo (http://microformats.org/wiki/geo)

Evento geolocalizado expresado con hCalendar

```
<div id="empresadigital: event1" class="vevent">
  <h1><span class="summary">OSGI: una Plataforma Modular y Ligera para
   Construir Aplicaciones basadas en Servicios</span></h1>
  <h2><abbr title="20080401T0900" class="dtstart">22 Mayo del 2008 9am a
   </abbr><abbr title="20080401T1400" class="dtend">2pm</abbr> en <span
   class="location">Parque Tecnológico de Álava, Miñano</span></h2>
  <abbr class="geo" title="42.883; -2.760"></abbr>
  <p class="description">OSGi es una de las tecnologías de software
   distribuido que más impacto está causando últimamente. Ven a este curso
   y aprende que hay detrás de OSGi.<a
   href="http://www.empresadigital.net/osgi" class="url">Pulsa</a> para
   obtener más información.</p>
  <p><span class="organizer vcard">Organizado por</span>
    <a href="http://www.arabadigitala.com" class="url"><span class="fn
   org">Araba Enpresa Digitala</span></a>. Contáctanos en <a class="email"
   href="mailto:events@enpresadigitala.net"> events@enpresadigitala.net</a>
  </p>
  <p><span class="logo" content="images/logo_ARABA.gif"> <img
   src="images/logo_ARABA.gif"/></span></p>
</div>
```


Evento geolocalizado en RDFa

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!--<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML+RDFa 1.0//EN" "http://www.w3.org/MarkUp/DTD/xhtml-rdfa-1.dtd">-->
<html xmlns:cal="http://www.w3.org/2002/12/cal/ical#"
      xmlns:contact="http://www.w3.org/2001/vcard-rdf/3.0#"
       xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos#">
    <head>
          <title>Eventos de la Facultad de Ingenieria de la Universidad de Deusto</title>
    </head>
    <body>
          <p about="http://www.deusto.es/events/event1" instanceof="cal:Vevent">
                Charla técnica:
                <span property="cal:summary">
                      Web con minusculas
                </span>
                a celebrarse el dia
                <span property="cal:dtstart" content="20080412T1600-0500">
                      12 de Marzo a las 4pm.
                </span>
                en Auditorio Principal Universidad de Deusto
                <span href="#p1" rel="geo:Point">
                      <span about="#p1">
                            <span property="geo:lat">43.270737</span>
                            <span property="geo:long">-2.939637</span>
                      </span>
                </span>
                organizado por
                <span href="http://www.deusto.es/staff/dipina" rel="cal:organizer">Diego Lopez de
    Ipina</span>
          </p>
          ...
```

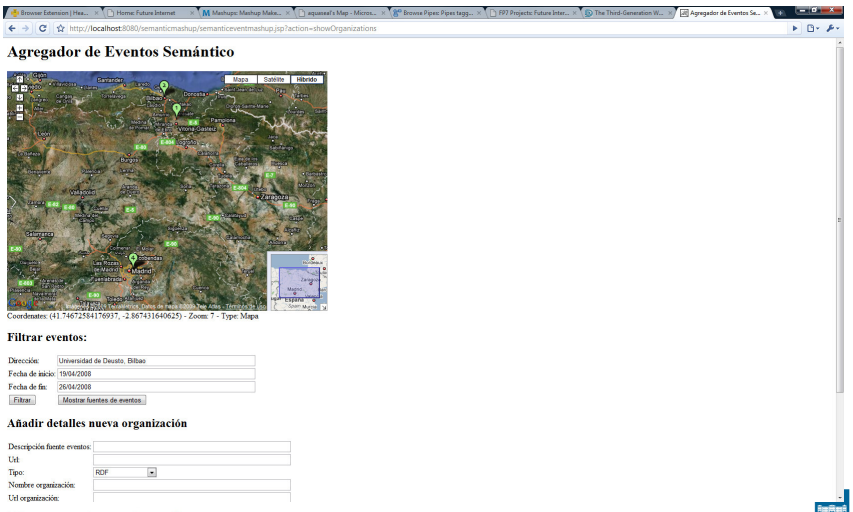
Evento geolocalizado expresado en RDF

```
<?xml version="1.0" encoding="iso-8859-1"?>
<rdf:RDF xmlns:cal=http://www.w3.org/2002/12/cal/icaltzd#
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos#"
  xmlns:vcard="http://www.w3.org/2001/vcard-rdf/3.0#">
  <rdf:Description rdf:about="http://www.revistasprofesionales.com/solop#event2">
    <rdf:type rdf:resource="http://www.w3.org/2002/12/cal/icaltzd#Vevent"/>
    <cal:organizer rdf:resource="http://www.revistasprofesionales.com/solop"/>
    <cal:dtend rdf:datatype="http://www.w3.org/2001/XMLSchema#date">2008-05-27</cal:dtend>
    <cal:summary rdf:datatype="http://www.w3.org/1999/02/22-rdf-syntax-ns#XMLLiteral">Charla sobre Web 3.0</cal:summary>
    <cal:description rdf:datatype="http://www.w3.org/1999/02/22-rdf-syntax-ns#XMLLiteral">Ven a ver lo último en la sinergia entre Web 2.0
        y Web Semántica (GRDDL, RDFa, microformátos)</cal:description>
    <cal:url rdf:resource="http://www.revistasprofesionales.com/solop/charla2"/>
    <cal:dtstart rdf:datatype="http://www.w3.org/2001/XMLSchema#date">2008-05-27</cal:dtstart>
    <cal:location rdf:datatype="http://www.w3.org/1999/02/22-rdf-syntax-ns#XMLLiteral">Madrid, España</cal:location>
     <geo:Point>
      <rdf:Description rdf:about="p1">
        <geo:lat rdf:parseType="Literal">40.437</geo:lat>
           <geo:long rdf:parseType="Literal">-3.625</geo:long>
      </rdf:Description>
     </geo:Point>
  </rdf:Description>
...
</rdf:RDF>
```


Consulta que extrae eventos agregados

```
PREFIX cal: <http://www.w3.org/2002/12/cal/icaltzd#>
PREFIX contact: <http://www.w3.org/2001/vcard-rdf/3.0#>
PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
PREFIX xsd:    <http://www.w3.org/2001/XMLSchema#>
SELECT DISTINCT ?summary ?description ?dtstart ?dtend ?lat ?long ?org ?url ?email ?logo
WHERE {
  ?event cal:summary ?summary.
  ?event cal:dtstart ?dtstart.
  OPTIONAL { ?event cal:organizer ?organizer.
             ?organizer contact:fn ?org.
             ?organizer contact:url ?url.
             ?organizer contact:email ?email.
             OPTIONAL { ?organizer contact:logo ?logo. } }
  OPTIONAL { ?event geo:Point ?point.
             ?point geo:lat ?lat.
             ?point geo:long ?long. }
  OPTIONAL { ?event cal:geo ?loc.
             ?loc <http://www.w3.org/1999/02/22-rdf-syntax-ns#first> ?lat.
             ?loc <http://www.w3.org/1999/02/22-rdf-syntax-ns#rest> ?restgeo.
             ?restgeo <http://www.w3.org/1999/02/22-rdf-syntax-ns#first> ?long. }
  OPTIONAL { ?event cal:dtend ?dtend. }
  OPTIONAL { ?event cal:description ?description. }
  FILTER ( xsd:dateTime(?dtstart) >= xsd:dateTime("2008-04-14T00:00:00Z") )
  FILTER ( xsd:dateTime(?dtend) <= xsd:dateTime("2008-04-21T00:00:00Z") )
} ORDER BY ?event;
```

Semantic Mashup



References

- European Future Internet Portal, http://www.future-internet.eu/
- The Future of the Internet, Bled 31 March 2008, ftp://ftp.cordis.europa.eu/pub/fp7/ict/docs/ch1-g940-280-future-internet-ld_en.pdf
- Future Internet: The Cross-ETP Vision Document, Version 1, 8. January, 2009, http://www.future-internet.eu/fileadmin/documents/reports/Cross-ETPs_FI_Vision_Document_v1_0.pdf
- How Web 3.0 Will Work, http://computer.howstuffworks.com/web-30.htm
- Three New Search Services: Wolfram|Alpha, Microsoft Bing, Google Squared, http://dltj.org/article/alpha-bing-squared/
- Bing Search Engine, http://en.wikipedia.org/wiki/Bing_(search_engine)

References

- Cloud Computing – Disruptive Innovation & Enabling Technology, http://blog.gogrid.com/2008/08/20/presentation-cloud-computing-disruptive-innovation-enabling-technology/
- Web Applications 1.0, http://www.whatwg.org/specs/web-apps/2005-09-01/
- The Internet of Things
  - http://www.itu.int/osg/spu/publications/internetofthings/InternetofThings_summary.pdf
- Enterprise 2.0
  - http://blogs.zdnet.com/Hinchcliffe/
- Programmable Web
  - http://www.programmableweb.com

Deusto tech
Tecnológico Fundación Deusto
Teknologikoa Deustu Fundazioa
Deusto Technology Foundation

Towards Future Internet: Web 3.0, Internet of Services & Internet of Things

8 de Julio 2009, 11:30-13:30

Sala de Videoconferencias, ESIDE

Dr. Diego López de Ipiña (y algunos de mis doctorandos)
dipina@eside.deusto.es