Protocols to the Cloud

http://git.eti.pg.gda.pl/intel-grant/pliki/esa/Embedded_Systems_Architecture_P7.pdf

The concept of the Cloud

Cloud computing - an abstract concept in a distributed data processing

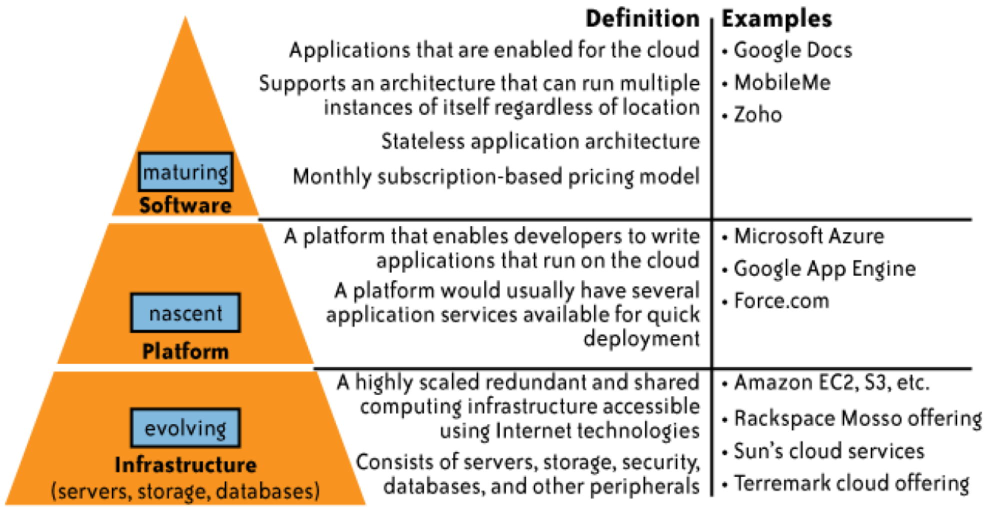
Features:

- multitenancy (shared resources)
- massive scalability (scaling operations for multiple systems and resource utilization)
- flexibility (increase or decrease computing resource „on demand”)
- „pay as you go” - the user pays for the resources (and time) actually used
- self-provisioning - the user decides about the resources they need

SPI framework

SPI - acronym for three basic types of services in the cloud:

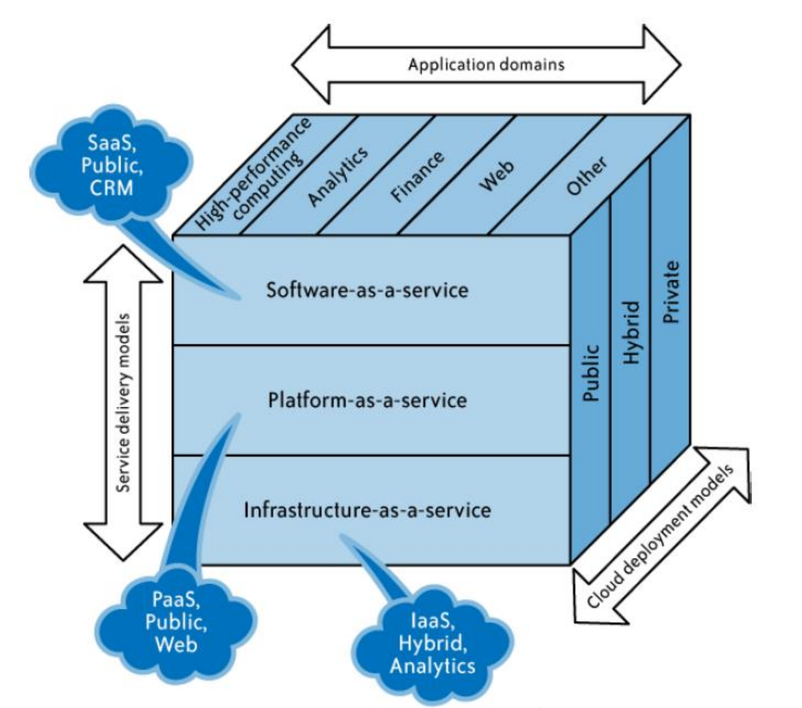
- software-as-a-service (SaaS),
- platform-as-a-service (PaaS),
- infrastructure-as-a-service (IaaS).



	Definition	Examples
maturing Software	Applications that are enabled for the cloud Supports an architecture that can run multiple instances of itself regardless of location Stateless application architecture Monthly subscription-based pricing model	• Google Docs • MobileMe • Zoho
nascent Platform	A platform that enables developers to write applications that run on the cloud A platform would usually have several application services available for quick deployment	• Microsoft Azure • Google App Engine • Force.com
evolving Infrastructure (servers, storage, databases)	A highly scaled redundant and shared computing infrastructure accessible using Internet technologies Consists of servers, storage, security, databases, and other peripherals	• Amazon EC2, S3, etc. • Rackspace Mosso offering • Sun's cloud services • Terremark cloud offering

While cloud-based software services are maturing, cloud platform and infrastructure offerings are still in their early stages

SPI framework



Cloud Security and Privacy, Shahed Latif, Subra Kumaraswamy, Tim Mather. Publisher, O'Reilly Media, Inc. Release Date: September 2009

Software-As-a-Service

SaaS model:

- user rents software for use on a subscription or pay-per-use model (an operational expense, known as OpEx)
- in some cases, the service is free for limited use
- user works with application using any device that can authenticate (a browser)

Traditional approach:

- customer loads the software onto his own hardware in return for a license fee (a capital expense, known as CapEx)
- support fee
- user takes care of the OS compatibility and compliance with license conditions

Platform-As-a-Service

User (programmer) uses a software development in the cloud

- tools
- standards
- channels of distribution and payment

Advantages:

- rapid dissemination of products, low cost of application development and market entry.
- developer builds applications without installing tools on their own machines
- the programmer does not need to be an expert!

PaaS is a variety of SaaS - borrowed service environment is creating applications (and the predefined blocks of code from which the user builds custom applications)

Infrastructure-As-a-Service

IaaS model:

- user pays for the amount of computing power, disk space and other resources actually consumed
- user has to care about the details of infrastructure: location, data management, scale, security, backup.

Features IaaS model:

- scalability - resources used may vary depending on current requirements, almost in "real time"
- pay-as-you-go: the user buys the exact amount of infrastructure, which currently needs
- best-of-breed: access to the best technology for a fraction of the cost

Cloud - advantages

- Low cost of investment and maintenance of the product (including - the end of production)
- Low cost of poor estimation of the resources required in the phase of development, testing and production
- Open standards. Open source = software subject to public license, allowing both to use and to modify the software. The consequence is the continuous development of open source solutions.
- Stability. Owners of the clouds are investing large amounts of human labor and money in the development of highly stable environment. The user can trust that the cloud has:
  - limited number of errors,
  - high resistance thanks to clusters,
  - ability to continuously develop solutions to ensure stability.

Who benefits from cloud computing?

Consumers

- e-mail, pictures, music
- information (personal) social networks
- maps, GPS
- websites
- cooperation (eg. Google Sites)

Business users

- development of business applications
- development of websites (advertising)
- sales through dedicated websites (eBay)
- advertising on search engines
- financial management in online banking
- office tools

Who benefits from cloud computing?

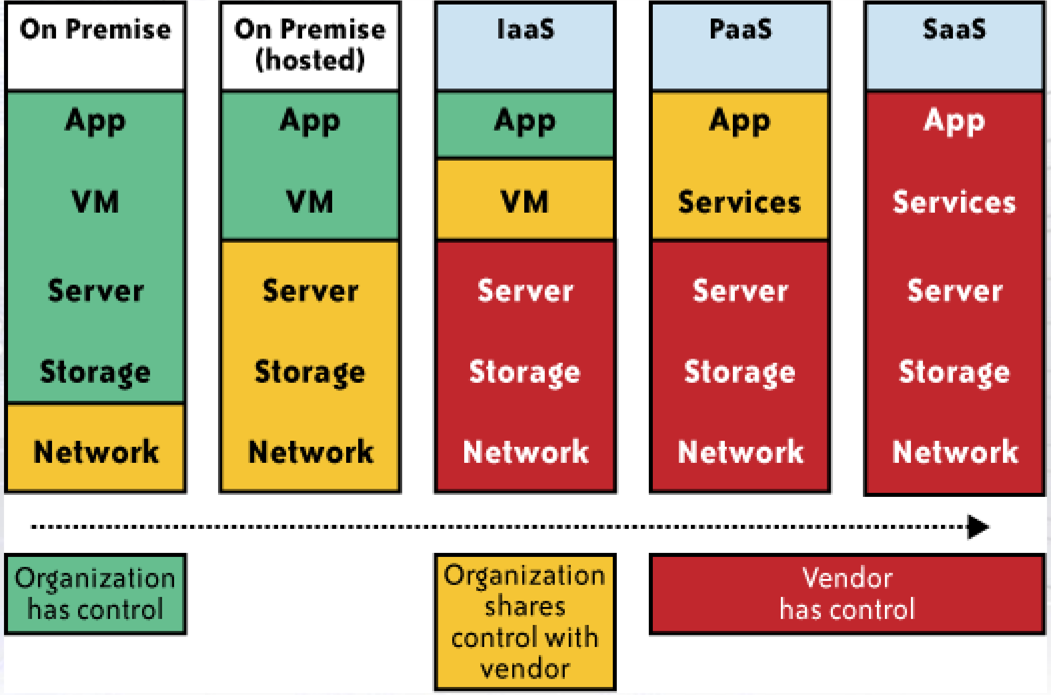
Start-ups, small businesses

- outsourcing of IT infrastructure that is crucial for product launch
- scalability
- in many cases - pioneering use of cloud computing

Large companies / corporations

- tools to support employee productivity (search engine knowledge, "travel service")
- tools to support HR and evaluation of employees
- tools to support key sectors: sales, document management, raw material sourcing, logistics (sensitive data!)

Property and control in a Cloud



On Premise	On Premise (hosted)	IaaS	PaaS	SaaS
🟩 App	🟩 App	🟩 App	🟨 App	🟥 App
🟩 VM	🟩 VM	🟨 VM	🟨 Services	🟥 Services
🟩 Server	🟨 Server	🟥 Server	🟥 Server	🟥 Server
🟩 Storage	🟨 Storage	🟥 Storage	🟥 Storage	🟥 Storage
🟨 Network	🟨 Network	🟥 Network	🟥 Network	🟥 Network

🟩 Organization has control

🟨 Organization shares control with vendor

🟥 Vendor has control

Cloud Security and Privacy, Shahed Latif, Subra Kumaraswamy, Tim Mather. Publisher, O'Reilly Media, Inc. Release Date: September 2009

Cloud example - Google

Google App Engine - platform-as-a-Service (PaaS)

- building and hosting web applications on Google's infrastructure
- supported programming languages: Python and Java.
- the service is free until the application is not using Google resources above the limit (volume data throughput, CPU cycles)

Google Apps - Software-as-a-service (SaaS)

- use of office applications: Gmail, Google Calendar, Talk, Docs and others.
- standard version is free (like Gmail)

Google Cloud - use cases

- Communication
  - description: corporate mail and calendar without creating infrastructure within the company
  - service: Gmail, Google Calendar
- Safe e-mail
  - description: securing an existing mail system without creating infrastructure within the company; filtering spam, viruses and other threats
  - service: Google Email Security
- E-mail storage and legal discovery for existing email systems
  - description: organizations can leverage Google Apps for managing email retention with a searchable archive so that they can locate email quickly in the event of legal discovery without the investment and maintenance of hardware and software.
  - service: Google Email Archiving and Discovery

Google Cloud - use cases

- Cooperation
  - Description: common access to tools and office documents without installing additional software on local machines or servers
  - service: Google Docs, Google Drive, Google Sites

- Creating applications
  - description: application development in Java or Python without investing in infrastructure (software and hardware) on local machines.
  - service: Google App Engine

Katedra Inżynierii Komputerowej

intel

Sponsor specjalności

SOA

SOA (Service-Oriented Architecture)

- Concept of creating information systems, in which the main emphasis is put on the definition of services that will meet user's requirements.
- Includes a set of organizational and technical methods aimed at linking the business side of the organization of its IT resources.
- Business logic is broken between multiple distributed components, services, coordinated by a central control application.
- Service components are implemented and made available by independent entities, called service providers (Service Provider).
- The communication between the control application and the service components is done via the Internet.

Web Services

Web Service

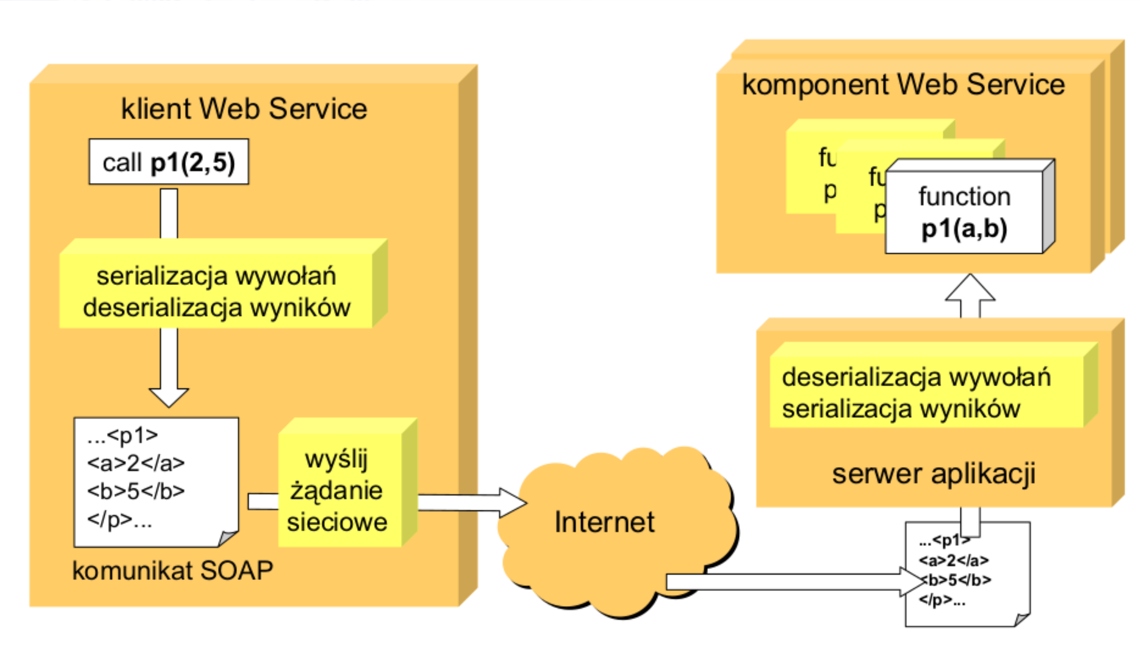
- Implemented programmatically service provided through the telecommunications network, including the network, particularly on the Internet.
- Software component, independent of the hardware platform and implementation, providing specific functionality.
- Communication between applications in heterogeneous and distributed environments. Web services offer - to varying degrees - unify communications regardless of the technologies used in the application.
- Data usually transmitted via HTTP and using XML.

Web Services

Methods of implementation:

- Defined by the services description language (WSDL)
- WSDL (Web Service Description Language) contains all the information needed to use the web service, including a list of available operations, their parameters and data types used
- Published using a standard mechanism, for example UDDI
- Called remotely by a defined interface
- Part of other online services

Web Services

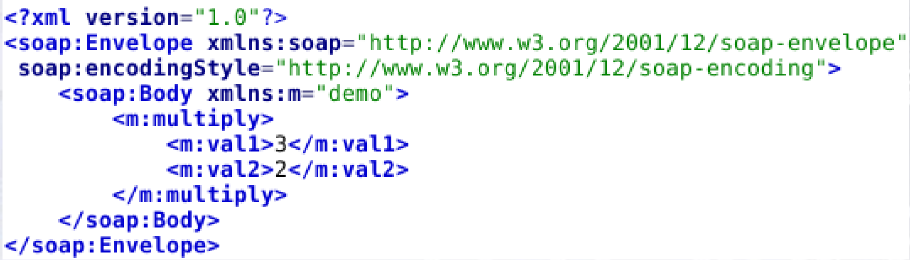

klient Web Service
call p1(2,5)
serializacja wywołań deserializacja wyników
...<p1> <a>2</a> <b>5</b> </p>...
wyślij żądanie sieciowe
komunikat SOAP
Internet
komponent Web Service
fu
p
fu
p
function p1(a,b)
deserializacja wywołań serializacja wyników
serwer aplikacji
...<p1> <a>2</a> <b>5</b> </p>...

http://www.cs.put.poznan.pl/mzakrzewicz/pubs/ploug06ws.pdf

SOAP

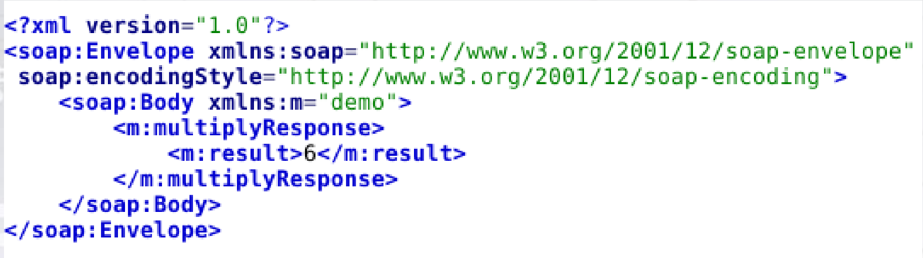
SOAP (Simple Object Access Protocol) - object-oriented communication protocol using XML to encode requests and (usually) HTTP to transfer them (the POST method)

Client request



```xml
<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
 soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
    <soap:Body xmlns:m="demo">
        <m:multiply>
            <m:val1>3</m:val1>
            <m:val2>2</m:val2>
        </m:multiply>
    </soap:Body>
</soap:Envelope>
```

Response

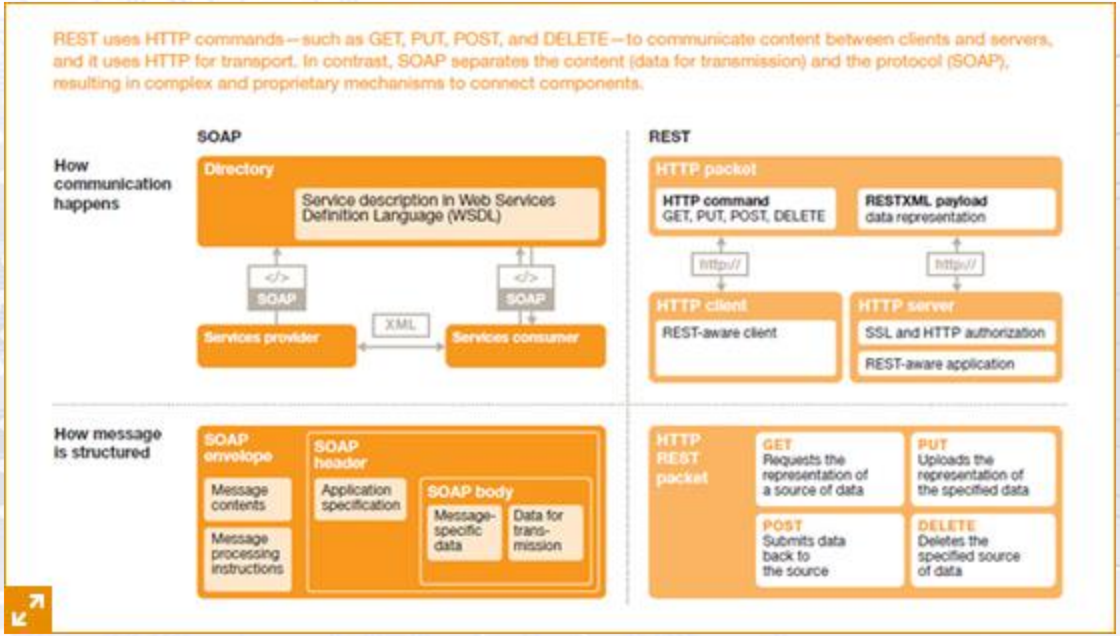


```xml
<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
 soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
    <soap:Body xmlns:m="demo">
        <m:multiplyResponse>
            <m:result>6</m:result>
        </m:multiplyResponse>
    </soap:Body>
</soap:Envelope>
```

REST

- REST (REpresentational State Transfer) - style of software architecture; a set of recommendations and "best practices" for creating scalable network services. Alternative to SOAP.
- Presented in 2000 by one of the authors of HTTP - Roy Fielding
- REST = resource (object / resource, URL) + representation (URI - Uniform Resource Identifier)
- Each resource (or any subset) may be on another host
- A resource can be represented as a result of type XML, JSON, Image, and others.
- RESTful webservices (RESTful web API) - network services based on the HTTP protocol and the main principles of REST pattern.
- HTTP methods: GET, POST, PUT, DELETE

REST vs SOAP



REST uses HTTP commands—such as GET, PUT, POST, and DELETE—to communicate content between clients and servers, and it uses HTTP for transport. In contrast, SOAP separates the content (data for transmission) and the protocol (SOAP), resulting in complex and proprietary mechanisms to connect components.

SOAP
REST
How communication happens
Directory
Service description in Web Services Definition Language (WSDL)
</>
SOAP
</>
SOAP
Services provider
XML
Services consumer
HTTP packet
HTTP command
GET, PUT, POST, DELETE
RESTXML payload
data representation
http://
http://
HTTP client
REST-aware client
HTTP server
SSL and HTTP authorization
REST-aware application
How message is structured
SOAP envelope
Message contents
Message processing instructions
SOAP header
Application specification
SOAP body
Message-specific data
Data for trans-mission
HTTP REST packet
GET
Requests the representation of a source of data
PUT
Uploads the representation of the specified data
POST
Submits data back to the source
DELETE
Deletes the specified source of data

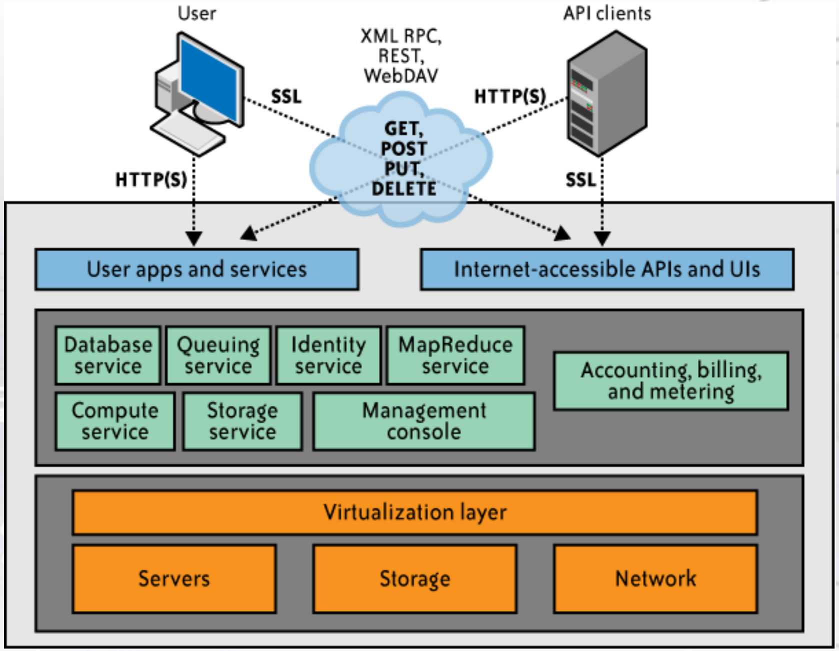
http://www.pwc.com/us/en/technology-forecast/2012/issue2/features/feature-consumerization-apis.jhtml

Elements of the Cloud

- Cloud access devices - home PCs, buiseness PCs, mobile devices, embedded systems)
- Internet browsers - access to information and aplications
- Broadband network
- Data centers and server farms
- Storage devices
- Virtualization techniques; subjects:
  - operating systems (VMware, Xen, DropBox, Docker),
  - data storage systems (NAS, SAN)
  - data bases and applications (Apache Tomcat, JBoss, Oracle App Server, WebSphere).

Elements of the Cloud

API (Application Programming Interface) - allows the user to program control of resources and services in the cloud



Cloud Security and Privacy, Shahed Latif, Subra Kumaraswamy, Tim Mather. Publisher, O'Reilly Media, Inc. Release Date: September 2009

API technologies

- Communication in REST architecture: HTTP GET, POST, PUT,and DELETE requests
- API defines:
  - Common behaviors that apply across all requests and responses
  - Resource models, which describe the JSON data structures used in requests and responses
  - Requests that may be sent a cloud resources, and the responses
  - expected
- Developers need to become familiar with specific APIs to deploy and manage software modules to the *aas platforms.

API technologies

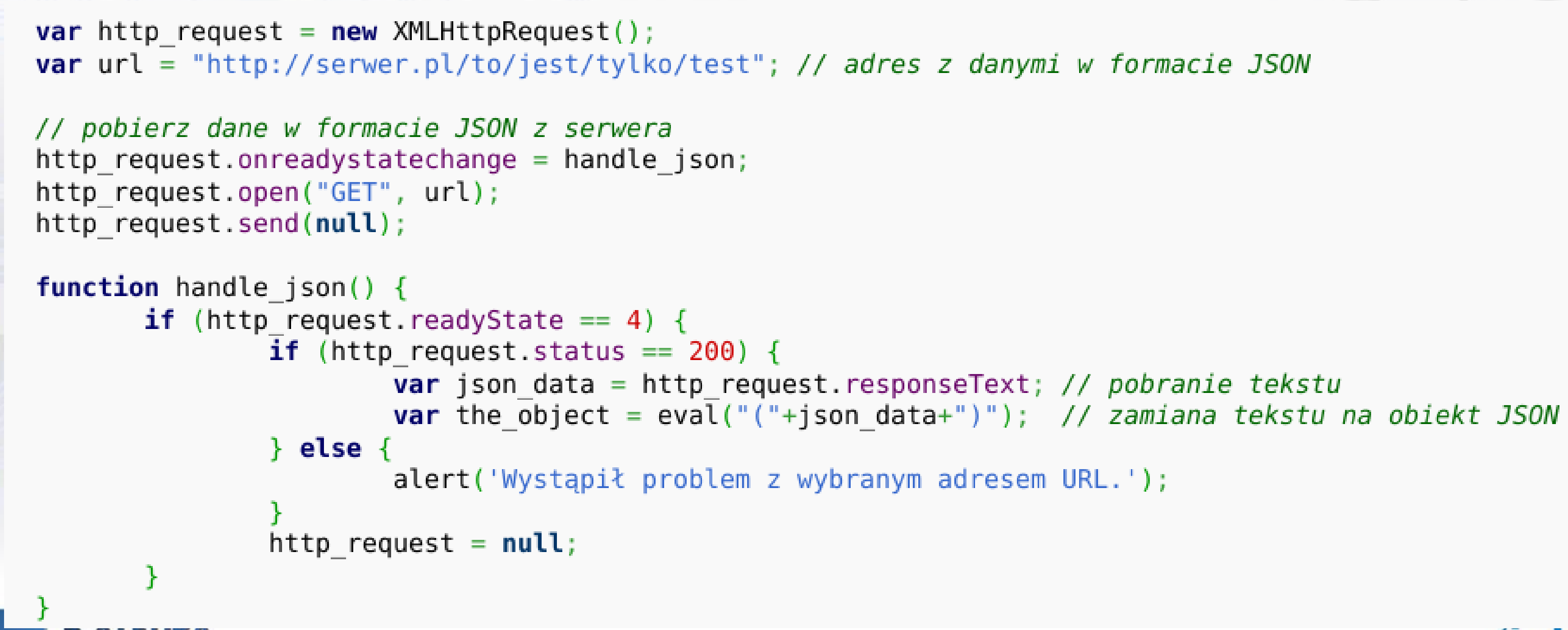
- Serialization - transformation of objects (instances of classes) into a stream of bytes, preserving the current state of the object. Serialized object can be:
  - recorded in a disk file,
  - sent to another process
  - sent to another computer over a network
  - stored in database
- Deserialization - reverse process to serialization.

API - JSON

JSON - JavaScript Object Notation - text data exchange format based on JavaScript.

- Despite its name JSON format is independent of any particular language.
- Supported by programming languages (additional packages or libraries): C, C ++, C #, Java, JavaScript, Perl, PHP, Python, Ruby
- Data in the JSON format are retrieved from the server as text (encoded by UTF-8) using the JavaScript XMLHttpRequest object, and then converted to an object.

API - JSON



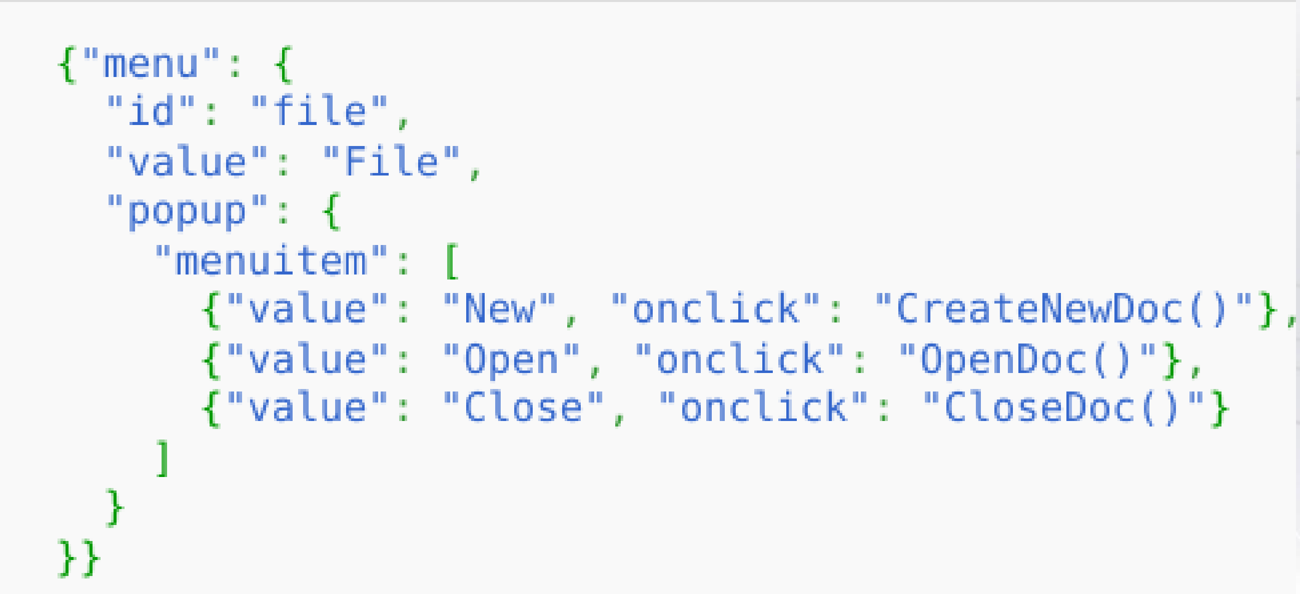
```
var http_request = new XMLHttpRequest();
var url = "http://serwer.pl/to/jest/tylko/test"; // adres z danymi w formacie JSON

// pobierz dane w formacie JSON z serwera
http_request.onreadystatechange = handle_json;
http_request.open("GET", url);
http_request.send(null);

function handle_json() {
        if (http_request.readyState == 4) {
                if (http_request.status == 200) {
                        var json_data = http_request.responseText; // pobranie tekstu
                        var the_object = eval("("+json_data+")");  // zamiana tekstu na obiekt JSON
                } else {
                        alert('Wystąpił problem z wybranym adresem URL.');
                }
                http_request = null;
        }
}
```

API - JSON

JSON message - literal of JavaScript object (hash table).

- All data are variables (they are not executable code) and the name of object properties are placed in quotes.
- Possible values:
  - String (string placed in quotation marks)
  - Number (double).
  - Constants: false, null, true,
  - Array composed of the above items
  - Object
- Objects and arrays can be arbitrarily nested.

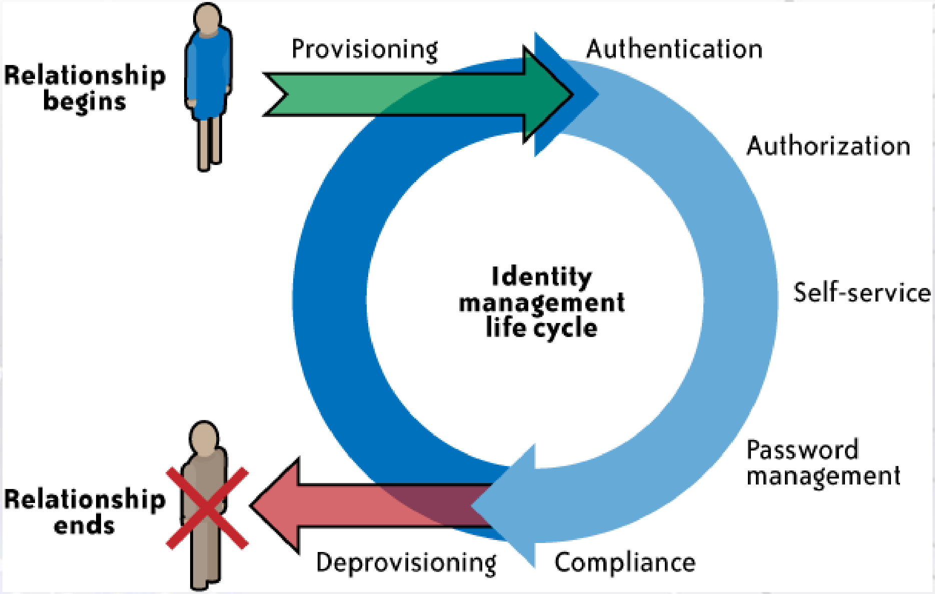
API - JSON



```
{"menu": {
  "id": "file",
  "value": "File",
  "popup": {
    "menuitem": [
      {"value": "New", "onclick": "CreateNewDoc()"},
      {"value": "Open", "onclick": "OpenDoc()"},
      {"value": "Close", "onclick": "CloseDoc()"}
    ]
  }
}}
```

Cloud Access - IAM

IAM = Identity and Access Management



Cloud Security and Privacy, Shahed Latif, Subra Kumaraswamy, Tim Mather. Publisher, O'Reilly Media, Inc. Release Date: September 2009

Authentication

Authentication management - activities for the effective governance and management of the process for determining that an entity is who or what it claims to be. Authentication stages:

- Identification - client declares its identity
  - telephone conversation with the bank's customer service center - client declares its data (the bank is relying party);
  - process of logging into the server - the user types the name (login) (the server is relying party);
  - browser connection to the SSL server - server presents an X.509 certificate containing its name (the browser is a relying party).

Authentication

- Authentication - the relying party applies the appropriate technique (authentication mechanism) in order to verify the declared earlier identity.
  - staff of the bank asks for a preset telephone code, date of birth, mother's maiden name; the sum of correct answers gives a high probability that the person is the one it claims to be;
  - server asks the user to enter a password (or indicate the key file) and verify its compliance with the preset value;
  - browser verifies the digital signature of the server certificate filed by a certificate authority that is trusted.
- Lightweight Directory Access Protocol (LDAP) - verifies the credentials of the client (client ID, unique user identifier assigned to him)

Authentication methods

- Paper documents - signatures, seals, initialed, watermark, notarial certificates;
- People and other living things - security biometric identification, password, smart card, biochip, token;
- Messages and electronic documents - a digital signature, message authentication code (message authentication code);
- Users of electronic communication - methods based on the proof of having a password (symmetric cryptography) or private key (asymmetric cryptography), one-time password.

Authentication methods

Functional classification of authentication methods

- something you know - information which is in the sole possession of the authorized entity, for example, a password or a private key;
- something you have - the object held by an authorized entity, for example, the key (to lock) or token (code generator);
- something you are - biometric methods.

Authorization

Authorization management - activities for the effective governance and management of the process for determining entitlement rights that decide what resources an entity is permitted to access in accordance with the organization's policies

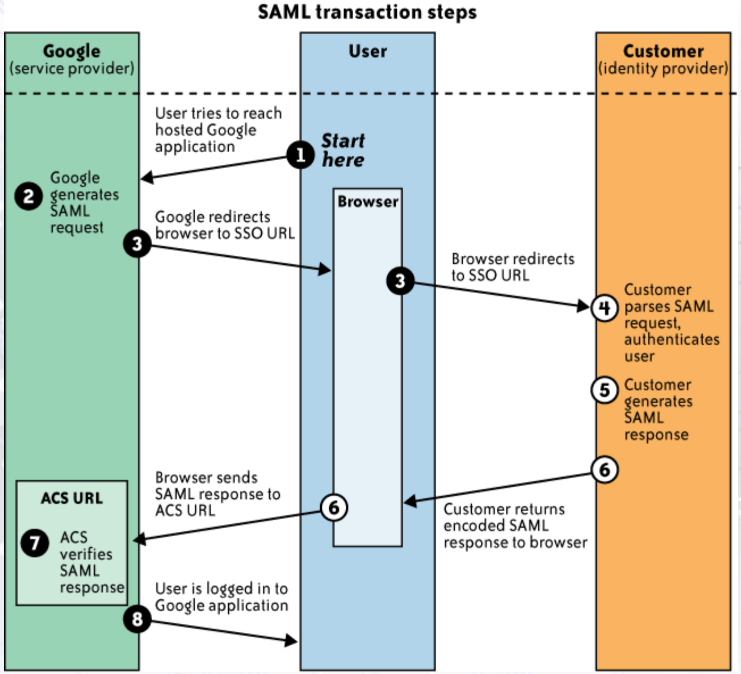
- Operating system checks the permissions of the authenticated user to the file based on its attributes in the fille system
- Internet banking user who is authenticated using the login and password, authorizes the transfer using electronic signature with one-time password
- Firewall simultaneously authenticates and authorizes access to the server port based on client IP address

IAM standards and specifications - SAML

SAML - Security Assertion Markup Language

- Protocol approved by OASIS (Organization for the Advancement of Structured Information Standards)
- Language for data exchange verification and authorization across domains
- Used to mediate authentication and automated transfer between systems and applications, the access rights of users
- Based on XML
- Clients - companies and organizations
- Solves the problem: how to avoid duplication of identity? (multiple log-ins to the web pages)

IAM standards and specifications - SAML


SAML transaction steps
Google
(service provider)
User
Customer
(identity provider)
User tries to reach hosted Google application
1
Start here
2
Google generates SAML request
Google redirects browser to SSO URL
3
Browser
3
Browser redirects to SSO URL
4
Customer parses SAML request, authenticates user
5
Customer generates SAML response
6
Customer returns encoded SAML response to browser
6
Browser sends SAML response to ACS URL
ACS URL
7
ACS verifies SAML response
User is logged in to Google application
8

Cloud Security and Privacy, Shahed Latif, Subra Kumaraswamy, Tim Mather. Publisher, O'Reilly Media, Inc. Release Date: September 2009

IAM standards and specifications - SAML

Example:

- User wants to connect with Gmail
- Google generates a SAML authentication request
- Google sends a redirect to the browser. URL contains encoded SAML authentication request
- Company's service LDP decodes the SAML request and authenticates the user (request for credential or checks for cookies).
- LDP generates a SAML response containing the encrypted user name (username) and sign it with public and private key.
- LDP passes the SAML response to the user's browser and redirects it to Google Assertion Consumer Service (ACS)
- Google ACS verifies the SAML response with a public key. If the verification is positive, the ACS redirects the user's browser to the destination address (Gmail)

IAM standards and specications - SPML

SPML - Service Provisioning Markup Language

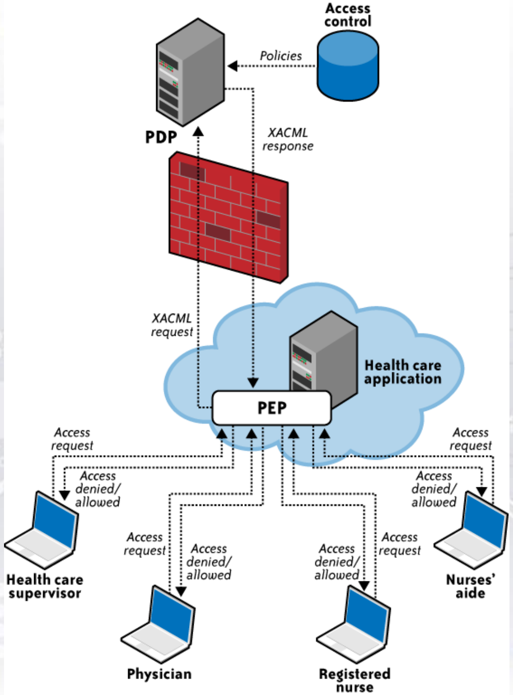
- Protocol approved by OASIS (Organization for the Advancement of Structured Information Standards)
- Used to automatically transfer user account information between the systems user account information
- Based on XML
- Clients - companies and organizations
- Solves the problem: how to transfer information about the user's account between the clouds?
- Automatically create new user accounts in "real time"

IAM standards and specifications - XACML

XACML - eXensible Access Control Markup Language

- Protocol approved by OASIS (Organization for the Advancement of Structured Information Standards)
- Based on XML
- Clients - companies and organizations
- Defines access control policy model and requests and responses protocol of communication between entity and decision-taking unit
- Enables uniform access control policies for different platforms and services
- Solves the problem: how to transfer information about user permissions between the clouds?

IAM standards and specifications - XACML



Cloud Security and Privacy, Shahed Latif, Subra Kumaraswamy, Tim Mather. Publisher, O'Reilly Media, Inc. Release Date: September 2009

IAM standards and specifications - XACML

- OAuth - standard of authentication and authorization
- solves the problem: how to authorize access of services X to data in service Y without revealing the credentials (for example username and password)?
- Allows users to share their private resources (eg. photos, videos, contacts) stored in one page to another page without having to go into the service of their credentials, usually providing the user name and token (OTP). Token allows them access to a specific page for specific resources and for a limited time.
- Allows the user to grant third parties access to information stored with another service provider without sharing permissions, or grant full access to the data.
- Open protocol.

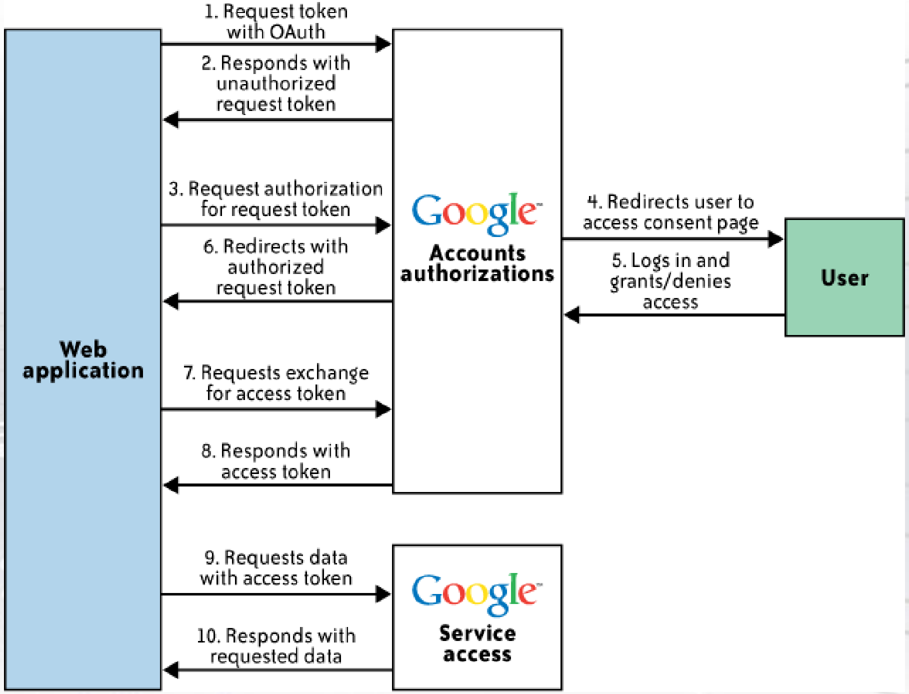
IAM standards and specifications - OpenID

OpenID - open standard for authentication and distribution of distributed identity of users in web-based services.

- solves the problem of distribution of the user identity between multiple web-based sites. Instead of creating separate accounts in each of the services, the user creates one OpenID account on the server, saving his personal information and obtaining OpenID.
- used by consumers; nearly doesn't exist in companies and institutions.

Google GData

Google has developed a hybrid version of OpenID and Oauth (GData APIs), reducing the number of actions needed for authentication and authorization


Web application
1. Request token with OAuth
2. Responds with unauthorized request token
3. Request authorization for request token
6. Redirects with authorized request token
7. Requests exchange for access token
8. Responds with access token
9. Requests data with access token
10. Responds with requested data
Google
Accounts authorizations
4. Redirects user to access consent page
5. Logs in and grants/denies access
User
Google
Service access

Cloud Security and Privacy, Shahed Latif, Subra Kumaraswamy, Tim Mather. Publisher, O'Reilly Media, Inc. Release Date: September 2009

Gdata-python-client

Google Data (Gdata) - Python library for communicatino with Google Drive

- Source:

https://code.google.com/p/gdata-python-client/downloads/list

- Installation:

```
cd gdata-2.0.18
sudo python setup.py install
```

- Example - working with Google Sheet

```
cd gdata-2.0.18/samples/spreadsheets
spreadsheetExample.py
```

Gdata-python-client

- Instruction for new client (project) registration in Google Developers Console:

https://developers.google.com/api-client-library/python/samples/samples

- Authorization

http://stackoverflow.com/questions/26925125/accesstokenrefresherror-google-spreadsheet-api-with-oauth-2-0-service-account-o?noredirect=1#comment42398164_26925125