Planning and Misc

Alexis Abbott Notes

ODM

STS

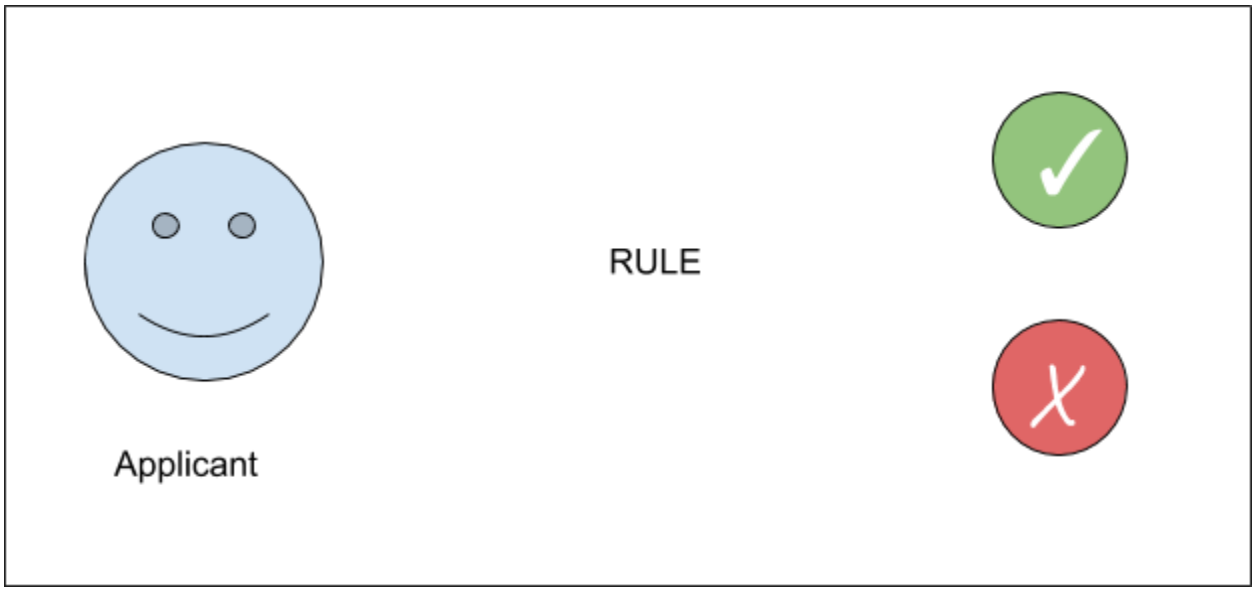
Table of Contents

ODM

Operational Decision Manager is about creating rules. Generally your BPM (Business Process Management) has policies on how to approach different processes. For example, a lender has an application for a loan:

IF salary > $50,000 AND contiguous years employed > 5 THEN loan = approved.

This is just one rule of possible hundreds or thousands a business could have.



Model Process

BPM will create schematics about full business processes. Usually centered around human actions, in other words, when processes aren't automated.

Often you'll have your business rules scattered throughout several components: software, subject matter experts, documentation, etc. The purpose of ODM is to have that information in one place. This achieves an SSOT (Single Source of Truth) and is very helpful in regard to auditing and logging. Very useful tool for a Business Analyst.

This also allows business and developers to work more seamlessly with each other as business can create the process rules in a codeless way, and the application can make calls to ODM without needing to hardcode it. You can test your rules in the IBM dashboard.

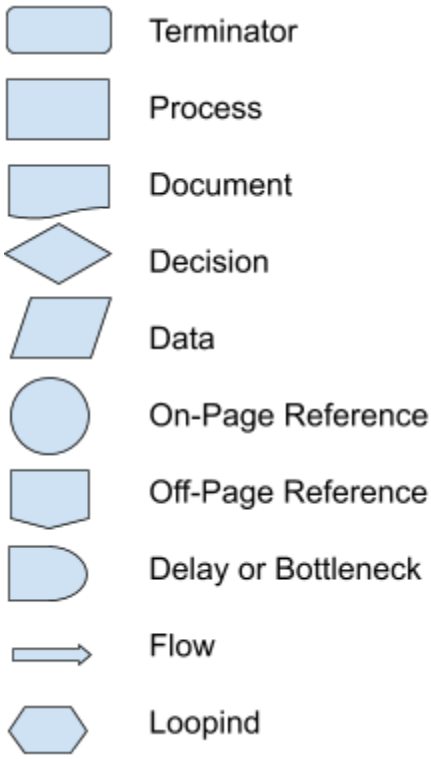
Planning

Flow Charts

Procedure: Finite sequence of well defined instructions, carried out in a finite amount of time.

Algorithm: A step-by-step procedure to solve a given problem.

A flowchart is a type of diagram that represents an algorithm or process. They're graphical tools to design the step by step process of an application. There are common shapes with standard meanings:



Terminator: Start or stop of a program.
Process: Indicates a particular program.
Document: Represents a print out.
Decision: Decision or branching point. Lines indicate possible scenarios or sub processes.
Data: Information entering or leaving the system. An input could be a shopping cart order. Output could be a product to be delivered.
On-Page Reference: Would contain a letter. Indicates symbol matching on page.
Off-Page Reference: Contains a letter and indicates symbol mathcing on another page.

Reasons to use a flow chart:

- Clarifies complex processes
- Identifies steps without value
- Understanding of process, using this info to collect data, identify problems, focus discussions, and identify resources.
- Serves as basis for designing new processes.

outline

A way to roughly go over the classes and what should be in them - the methods and what they're trying to accomplish.

Example

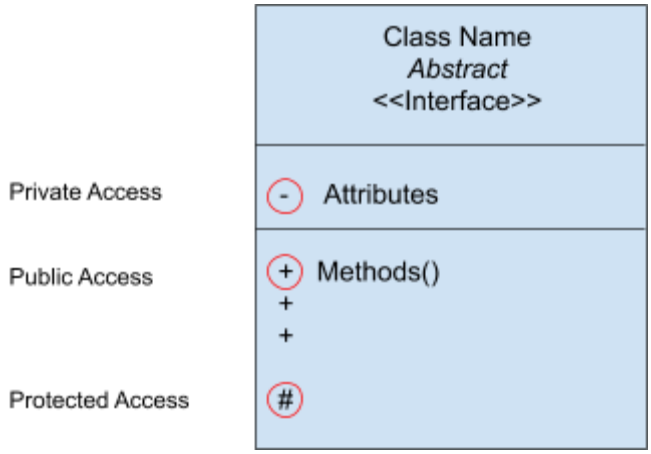
```
imports

public class WorkingProject{
  public void run(){
    //code that calls the rest.
  }

  public void behaviorDesc(){
    //process step 1
    //step 2
    //step 3
  }
}
```

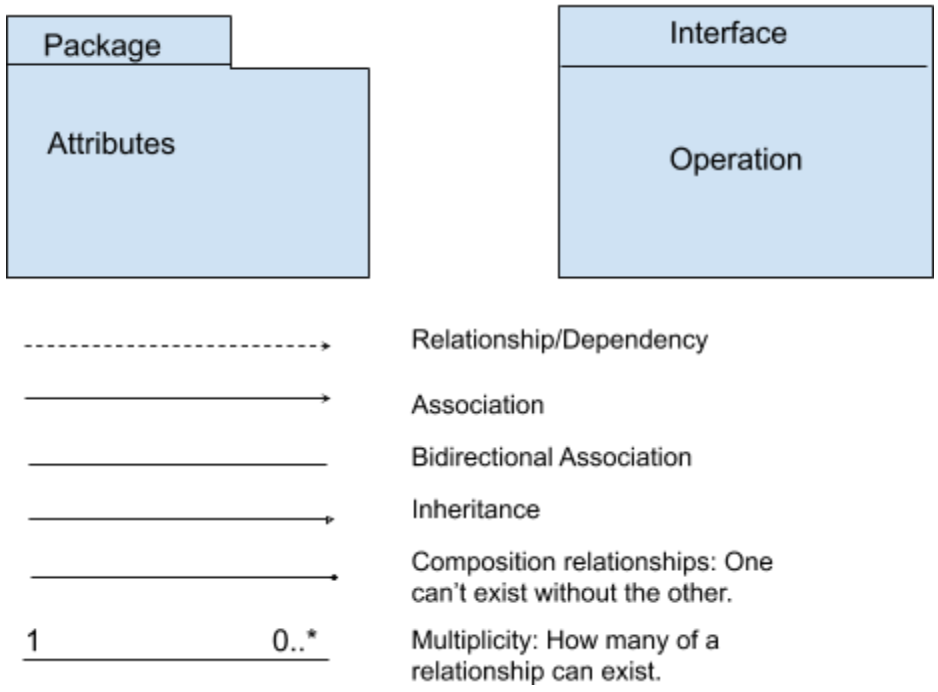
UML

Unified Modeling Language.



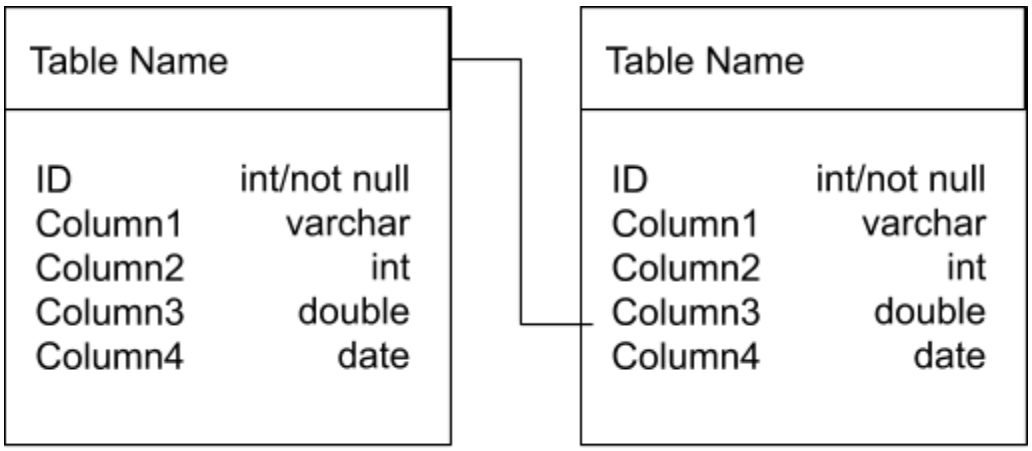
Omission of an access signifier means package-level visibility. If an operation or attribute is underlined, it denotes it's static.

```
method(datatype:param(s)):return datatype
```



ERD

Entity Relationship Diagram



User Stories

User Stories have 3 parts:

1. Persona
2. The feature
3. The need that feature satisfies.

As a (user) I want a (feature) so that I can (satisfy a need).

What is an Epic?

It's a very big User Story that likely contains smaller User Stories. From an Epic, you break it down into even smaller pieces.

One shouldn't deliver an incomplete feature, but can deliver a small part of a larger feature. Bring things down to the minimum viable product and add functionality from there.

The important thing about breaking up a design is to find the foundation. What will other pieces be built upon? Writing the most important stories first ensures the most important features get delivered.

With fixed release dates and shifting agendas, we can't assume everything will ship. Starting with foundational features maximizes value. You can gray out the extra features on a wireframe when adding to a user story to focus on the end goal. Grayed out areas can be notated with their User Story identifier (generally a number ID) to communicate they are separate.

Structure of a User Story:
1. Summary/Narrative
2. Wireframe
3. Acceptance Criteria

Acceptance Criteria

Write down what the expectation is for the User Story. This way the developer(s) can take a look and give the feature a reality check.

Sometimes minor tweaks will need to be made to stories. Instead of updating the wireframes, notate there's a tweak and have the policy if there's a conflict between story and wireframe, follow the story.

Keep things simple. Only add as much information as needed. Keep the language casual.

STS

TIPS

#1
You can right click in your class to open a subcontext menu.

Source > Generate Getters and Setters…

The wizard will walk you through to generate the boilerplate. There are also a lot of other options where the boilerplate can be added for you.

#2
Going to Preferences and Favorites lets you add standard packages. Content assist will propose static members even if the import is missing. Use the New Type button and then you can type in the class you want.

SHORTCUTS

sysout + space = System.out.println();

Rename multiple variable names: cmd + opt + r

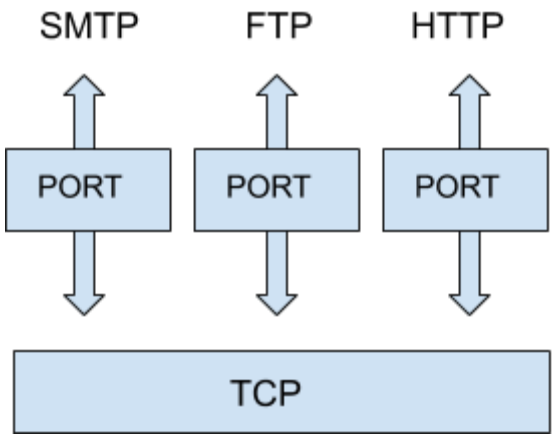
TCP/IP

Transmission Control Protocol

Maintains the state of connections.

A standard that defines how to establish and maintain a network conversation via which application programs can exchange data. TCP works with IP which defines how computers send packets to each other.

The two most common versions at the moment are IP4 and IP6. TCP works in layers. Transport (TCP, UDP) talks to the application layer through a port. Each port can work with a different protocol. This protocol is essentially how computers talk to each other.



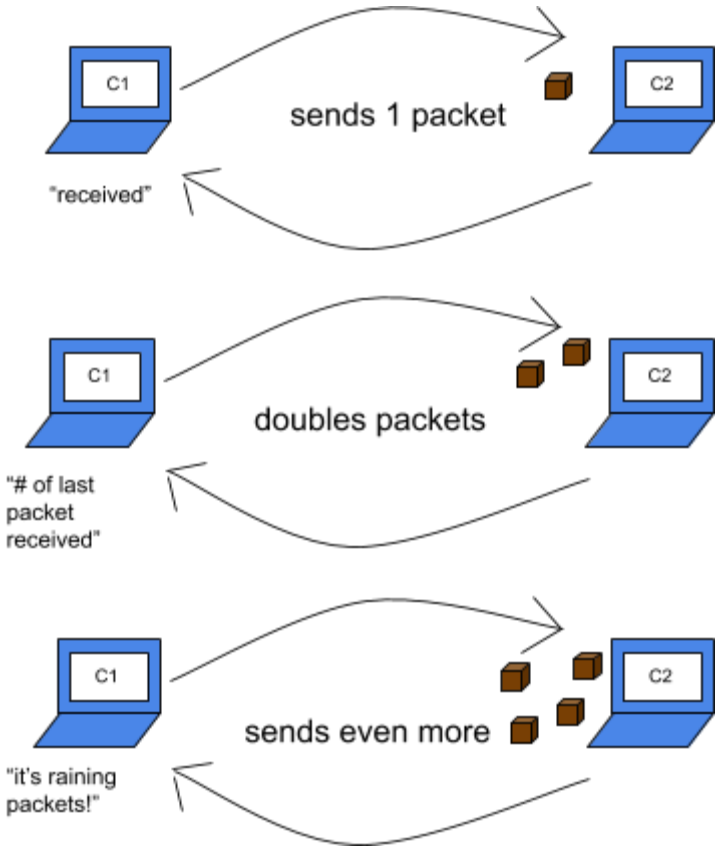
Once TCP gets data, it chops it up into packets. To put the packets back together, TCP puts a header on each packet with instructions for what order to reassemble them as well as error checking information.

A port is a number that uniquely identifies a socket owned by a process. Each process has to have a different socket. Ports are also an operating system resource. The OS directs communication to the correct process.

TCP Windowing Process

On a stable connection computer 1 (C1) will send a request to computer 2 (C2). C2 will send a packet as a response to test the connection. If C1 reports back it received the packet and gives the last packet number it received in order, C2 will double how many packets it sends. If things look good, C1 reports back with the correct number on the packets, C2 will again double and this will go on and on until either the transfer is complete, or a problem occurs.

If there's a problem with sending a group of packets, it will go back to sending just one and repeat the process.



IP - Internet Layer (Routing)

Attaches addresses to packets. Also allows for dividing huge networks into sub networks.

IP addresses have to be given to each computer. If you have a large network and you don't want all your computers talking to each other, you can create a subnet mask.

How IP addresses work:

4 octets - 10.0.1.0

Each section is an octet, it's made of 8 bytes, hence octet.

0	0	0	0	1	0	1	0
128	64	32	16	8	4	2	1

192.168.1.**10**

Examining this last octet, looking at the bytes, we add up the slots until we get the number we need. 8 + 2 = 10.

Network Layer

Checks for things like MAC addresses to send to the correct physical machine.

Common Protocols

A suite of protocols work together to exchange data over the internet. Below are the common ones. These are considered the application layer and utilize standard port numbers.

HTTP, FTP, SMTP

Ports/Port Ranges

Ports represent a logical connection to a particular piece of software running on a server.

0-65535 - available port range.

Default Gateway

This is the router for the subnet mask. If the IP can't find the machine it's trying to communicate with, it will go to the Default Gateway and ask. This process is often your physical router.

Subnet Mask

You can have multiple subnetworks on the same router.

2^n is how many subnets you can have where n is equal to how many bits you've assigned to the subnet. The subnet mask lets us know what portion of the IP is used for the subnet. Since at least one octet needs to be assigned to either the subnetwork or hosts, there are only three different types of masks possible. (4 octets in total.)

Possible Subnet Types

- A - 255.0.0.0
- B - 255.255.0.0
- C - 255.255.255.0

All of the bits in the octet add up to 255 and means in a way "on". 0 is no bits allocated and in a way means "off".

Type C Subnet

192	168	1	1
255	255	255	0

8 bits have been reserved for the hosts (machines) in this subnet. Or 2^8.

The formula for hosts is $2^n - 2$.

If you need a lot of subnets, Type C is the best result, but you won't be able to attach as many machines. If you need to make room for a lot of machines, including printers and general IoT, add more bits to the right. Type A might suit this need better.

On a class C subnet, if we wanted to further divide the hosts into different subnets, we add up the bits allocated.
255.255.255. 1 1 | 0 0 0 0 0 0
The marked bits mean we add them up - 128+64=192

Subnet mask would ultimately be:
255.255.255.192

Routers are needed between devices if they're on different subnets in order for them to talk to each other.

DHCP Server

Dynamically assigns IP addresses. An automatic way to assign IP addresses in the network. A DHCP server will be assigned a scope (range) of IP addresses it can give out. It will also assign a lease time for how long a computer can have that IP address. Halfway through the lease, the computer will request to extend it. If it can't contact the server, at the halfway point it will try again. It will continue to do this until it gets a response or the lease expires.

NAT

Network Address Translation
Every router has to have a unique IP address. NAT is at the router level. It's what routes the data packets to the specific computer that asked.

IPC Protocol

2 types of processes

- Independent
- Cooperating

Cooperative processes can synchronize their tasks using shared memory and message passing.

A socket is an IPC mechanism. It's an OS resource that serves to let two processes communicate with each other (process is a running program). The processes may or may not be on the same machine.

A process uses a socket like a file that it can write data which will get sent to the other side making it available to be read.

PID - Process Identification

UDP

User Diagram Protocol - considered unreliable. Used for time sensitive transmissions such as video playback or DNS lookups. Speeds up communication by not requiring a handshake, transferring data before the receiving party agrees to the communication. Creates an opening for exploitation. Doesn't have error checking.

Common uses for UDP:

- VOIP
- Online gaming
- DNS
- NTP

Port Scanning

TCP Scanning

Three-way handshake. Client sends a SYN (Synchronize Sequence Numbers) flag, the server responds with SYN-ACK, and the client responds with an ACK (Acknowledgment) flag to complete the handshake. If the server responds with RST (Reset the Connection) flag, the port is closed.

UDP Scanning

UDP is stateless and doesn't maintain the state of connection (unlike TCP). Scanner sends a UDP packet to the port and if it's closed, an ICMP packet is generated and sent to origin. If it doesn't, that port isn't open.

UDP scanning is unreliable because ICMP packets can be dropped by firewalls, generating false positives for port scanners.

Nmap

Versatile and comprehensive port scanner.

- Port Scanning
- Fingerprint OS
- Vulnerability Scanning

```
$ sudo nmap domain.com
```

To scan for UDP ports:

```
$ sudo nmap -sU domain.com
```

Nmap Flags:

- -p- Scan all 65535 ports
- -sT TCP connect scan
- -O Scan for OS
- -v Verbose scan
- -A Aggressive scan - everything
- -T[1-5] Set scanning speed
- -Pn In case the server blocks ping

Other Scanning Apps

Netcat
UnicornScan

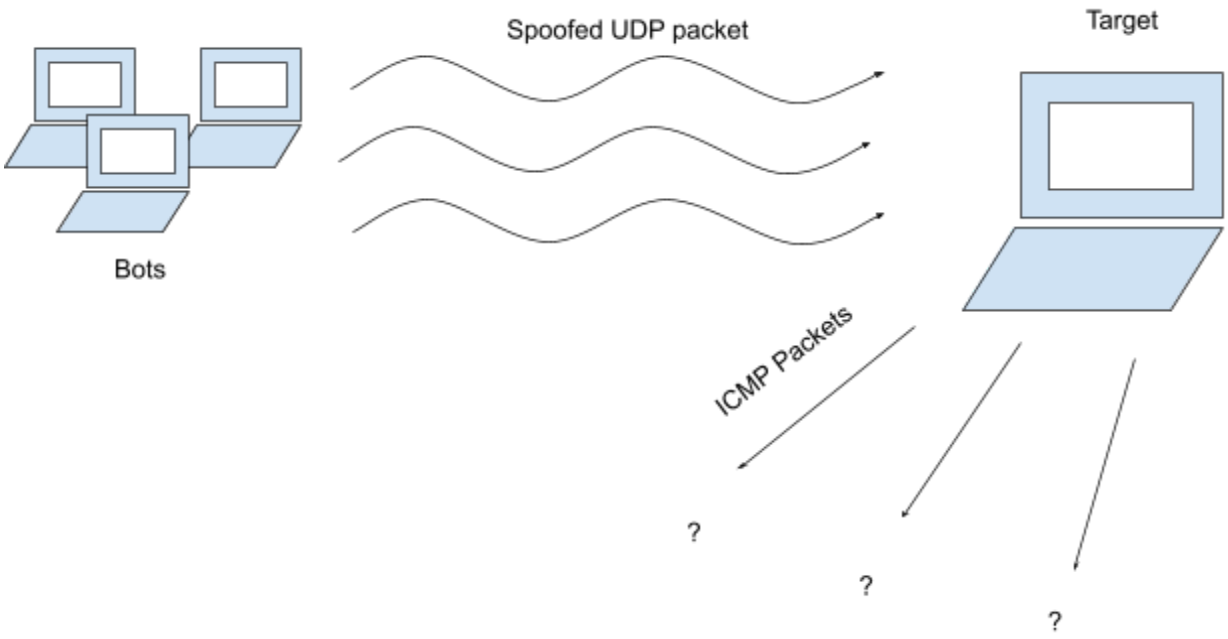
Firewall

A firewall is a filter. It will check its own table for rules to see if it will allow packets through.

A port being “stealthed” simply means incoming packets requesting to connect to that port are silently being dropped by the firewall before the OS sees them and has a chance to reply back. Normally the OS will send response messages such as “connection refused” or “connection accepted”. The firewall does not.

DDoS

DDoS attacks including DNS amplification and NTP (Network Time Protocol) amplification make use of vulnerable instances of these servers with the aim of flooding a target with UDP traffic.



OSI 7 Layer Network Model

7 - Application
6 - Presentation
5 - Session
4 - Transport
3 - Network
2 - Data Link
1 - Physical

7 - Application

What most users see, what they interact with directly.

6 - Presentation

Application format to network format or vice versa. The layer “presents” data for application or network. Decryption/Encryption

5 - Session

The connection between computers. Functions involve setup, coordination (wait time for responses), and termination.

4 - Transport

Coordinates data transfer between end systems and hosts. How much data to send, what rate, where it goes, etc. TCP or UDP

3 - Network

Router functionality and packet forwarding. (IP)

2 - Data Link

Node to node transfer and handles error correction from physical layer. Two sublayers:

- MAC - Media Access Control
- LLC - Logical Link Control

1 - Physical

Electrical and physical parts of the system. Includes cable type, radio frequency link (802.11 wireless), layout of pins, voltages, and more.

URI

Uniform Resource Identifier
Has two different identifiers: URL and URN
Both makeup the URI which get your full path.

URL

Uniform Resource Locator (where)
URLs tend to have the transmission protocol (https, ftp, smtp, etc…). It describes *how* to access it.

URN

Uniform Resource Name (what)
This tells us what the resource is. An example is an isbn number for a book:
urn:isbn:0-486-27557-4

SOAP

Simple Object Access Protocol
SOAP is XML based and sits on top of HTTP. It's a protocol that can use other TCP protocols (http, fpt, smtp, etc) to send XML responses from client requests. SOAP is largely just a data exchange in XML format.

CORS

Cross-Origin Resource Sharing is a W3C standard for allowing user agents to enable different-origin requests to take place in a secure way.

Simple method - case-sensitive:

- GET
- HEAD
- POST

Simple response header - case-insensitive

- Cache-Control
- Content-Language
- Content-Type
- Expires
- Last-Modified
- Pragma

Mapping web applications to distinct origins is vital for secure web applications. An origin is composed of a scheme (http), host name (domain), and port.

Preflight Requests

Asks the server if a method is allowed before using it. To get around access control blocking, you can proxy with a 3rd party server which will add appropriate response headers. However, it slows down the response time, impacting responsiveness.

Enabling CORS Support

When a browser receives a non-simple HTTP request, CORS requires the browser to send a preflight request to the server and wait for approval (or request for credentials) from the server before sending the actual request.

The preflight request appears to your API as an HTTP request that:

- Includes an Origin header
- Uses the OPTIONS method
- Includes the following headers:
    - Access-Control-Request-Method
    - Access-Control-Request-Headers

This means the API must have an OPTIONS method that can respond to the OPTIONS preflight request with at least the following response headers mandated by the Fetch standard.

- Access-Control-Allow-Methods

- Access-Control-Allow-Headers
- Access-Control-Allow-Origin

Enabling CORS depends on your API's integration type or server configuration.

Spring Boot:

```
@CrossOrigin(origins="http://alloweddomain.com")
```

There are other annotations for CORS support - look them up in the Spring Boot documentation.

By default all origins and GET, HEAD, and POST methods are allowed.

```
registry.addMapping("/end-point").allowedOrigins("http://alloweddomain.com")
```

RFC

Request for Comments is a standard for internet specifications, communication protocols, procedures, and events.

Normative - "Official" keywords that have understood meaning. Standard.
Non-normative - helper language used to clarify and explain.

ABNF

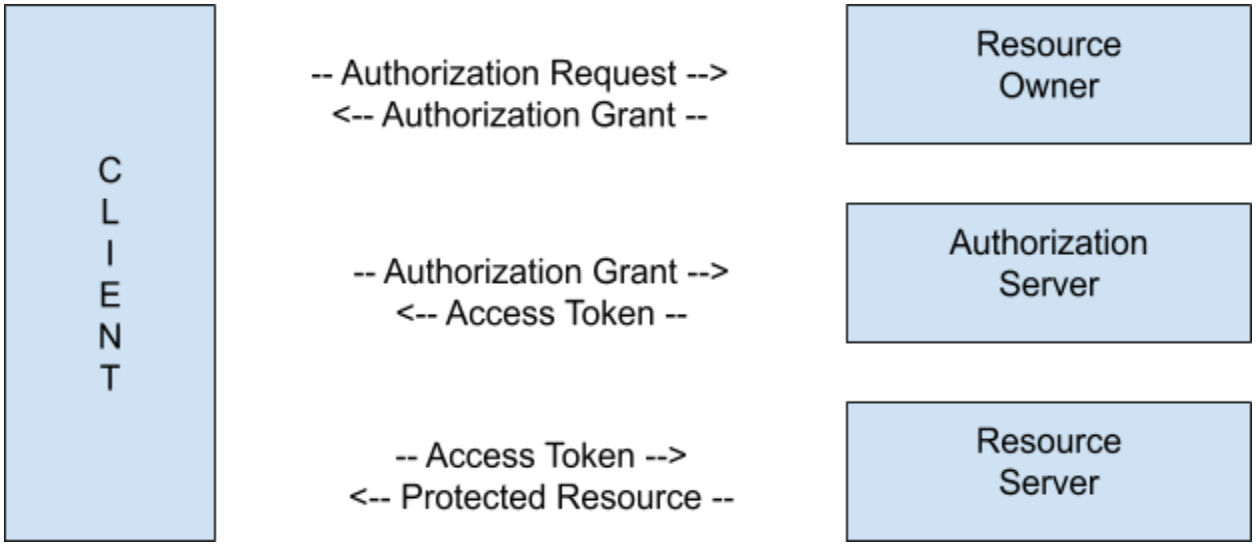
Augmented Backus-Naur is a metalanguage based on Backus-Naur Form (BNF). Formal system of language to be used as a bidirectional communications protocol. Serves as the definition language for IETF communication protocols.

OAUTH 2.0

In order to protect an owner's resources, the client obtains an access token - a string denoting a specific scope, lifetime, and other attributes. Access tokens are issued to third-party clients by an authorization server with the approval of the resource owner. Only works over HTTP.

Roles:

- Resource Owner - Entity capable of granting access to a protected resource. When it's a person it's referred to as an end-user.
- Resource Server - Server hosting protected resources, accepts and responds to requests with access tokens.
- Client - Application making protected resources requests on behalf of the resource owner and with its authorization.
- Authorization Server - server issuing access tokens to the client after authenticating the resource owner and obtaining authorization.



Authorization Grant:

- authorization code - authorization server acting as an intermediary to communicate between client and resource owner so authorization isn't exposed to either party.
- implicit - limits round trips and thus enhances performance, but at the cost of security. The client is issued an access token directly.
- resource owner password credentials - username and password can be used to directly obtain an access token. Requires a high degree of trust between resource owner and client.
- client credentials - when the authorization scope is limited to the protected resources under control of the client or resources previously arranged by the server.
- extensibility mechanism - defines additional types.

Tokens:

- Access Tokens - credentials used to obtain access protected resources. It's a string representing an authorization issued to the client.
- Refresh Tokens - credentials used to obtain access tokanes when the current one becomes invalid or expires. They are only ever used with authorization servers.

TLS has known security vulnerabilities.

Client Registration:
Usually involves an HTML form, can use other means to authorize the client.

Client Types:

- confidential - able to maintain the credentials securely.
- public - clients unable to maintain the confidentiality of their credentials.

These are defined by the authentication server's definition of secure authentication. The server should never assume anything about the client type.

Client Profiles:

- web application - confidential client running on a web server.
- user-agent-based - application public client where client code is downloaded from a web-server and executes within a user-agent.
- native application - a public client installed and executed on the device used by the resource owner.

Client Identifier:
Authorized server issues the registered client a client identifier - a unique string representing the registration information provided by the client. It is exposed to the resource owner and must not be used as the sole authentication.

Authorization Endpoint:
Used to interact with the resource owner and obtain an authorization grant. It must use TLS since user authentication is transmitted with clear-text credentials. It must also use the GET method and may support POST.

Redirection Endpoint:
After completing its interaction with the resource owner, the authorization server directs the resource owner's user-agent back to the client. The authorized server redirects the user-agent to the client's redirection endpoint previously established with the authorized server during the client registration process or when making the authorized request.

GIT