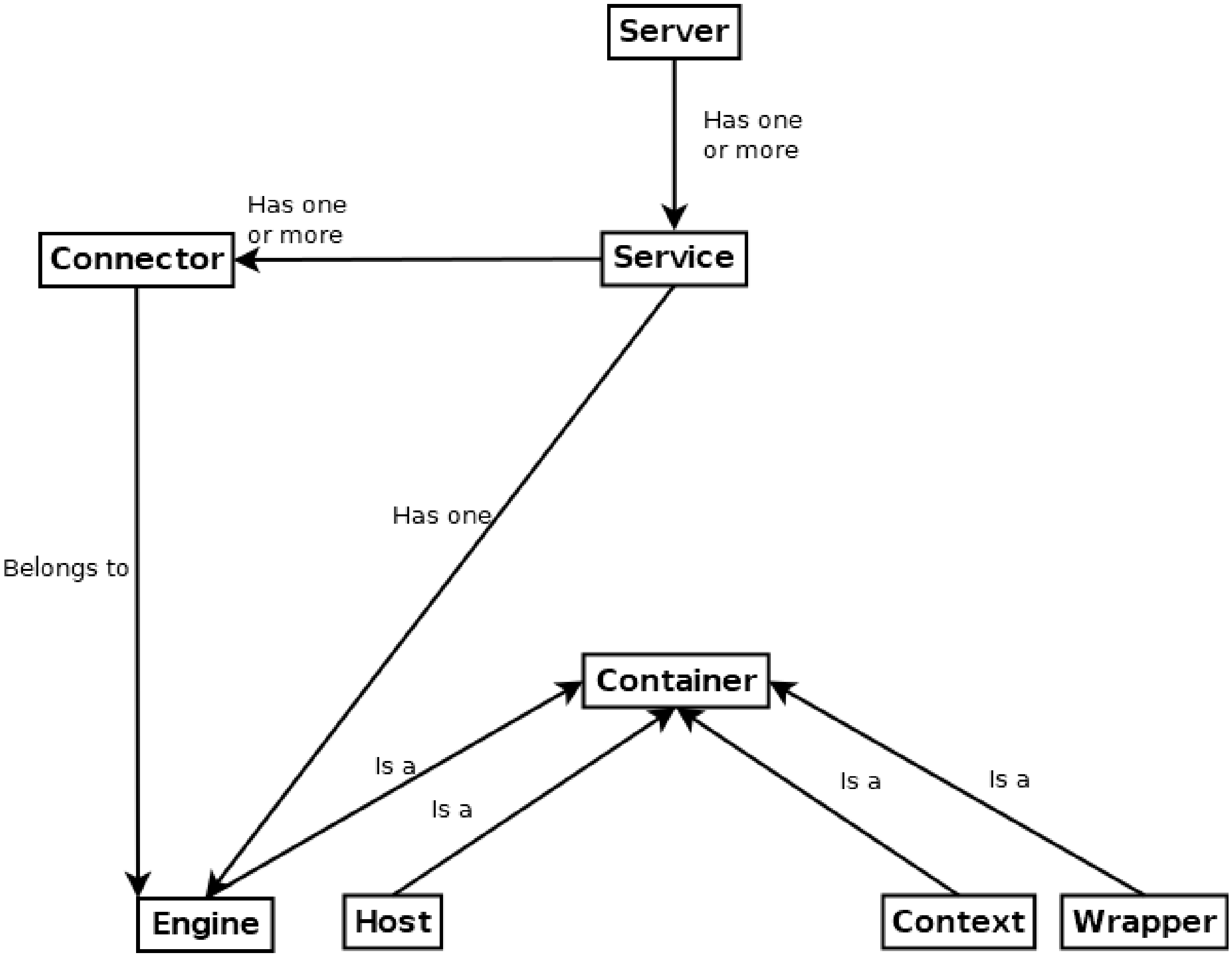
Rough Architecture



Server:

- Server Container
- Can have one or more services

Service:

- Lives inside a server
- Ties one or more connectors to one engine
- Belongs to one server
- Can have one or more connectors
- Can have one Engine

Engine:

- Processes requests for it’s services
- Receives and processes requests taken from the connectors
- Creates responses and handles them back to the connectors
- Belongs to one service
- Can have one or more hosts
- Has one or more connectors

Host:

- Association of network name (hs-mannheim.de) to Tomcat server
- Belongs to one engine
- Can have one or more contexts

Connector:

- Handles communications with clients
- HTTP-Connector for HTTP-traffic
- AJP-Connector to connect to e.g. HTTPD
- Belongs to one service
- Belongs to one engine

Context:

- Unique web application with unique path
- Belongs to one host

Interfaces and Classes

Server (org.apache.catalina):

- Interface
- Represents the Tomcat server
- Implementing class:
    - StandardServer (org.apache.catalina.core)

Service (org.apache.catalina):

- Interface
- Group of connectors that share a container (engine) for request processing
- Implementing class:
    - StandardService (org.apache.catalina.core)
        - Parent of Embedded
    - Embedded (org.apache.catalina.startup)
        - Convenience class to embed Catalina servlet inside another web application
        - After proper initialisation one can add connectors, engines, hosts and contexts on the
        - fly
    - Catalina (org.apache.catalina.startup)
        - Startup/Shutdown shell program for Catalina servlet

Engine (org.apache.catalina):

- Interface
- Useful for these scenarios:
    - Use of interceptors to see every single request processes
    - Running Catalina with a standalone HTTP-connector
- Implementing classes:
    - StandardEngine (org.apache.catalina.core)
- Container
    - Child-container is host-implementation

Host (org.apache.catalina.core):

- Virtual host in Catalina servlet engine
- Implementing classes:
    - StandardHost
- Container
    - Parent-container is engine-implementation
    - Child-container is context-implementation

Connector (org.apache.connector):

- Coyote connector

Context (org.apache.catalina):

- A web application in Catalina servlet engine
- Use of interceptor to see all requests for a context possible
- Implementing classes:
    - StandardContext (org.apache.catalina.core)
- Container:
    - Parent-Container = Host
    - Child-Container = Wrapper

NOTE: Engines and hosts aren't generally used when a Catalina is deployed which is connected to a Webserver like Apache because the connector then utilizes the webserver's facilities to determine the proper context for processing the request.

Wrapper (org.apache.catalina.core):

- Individual servlet definition
- Interceptor to see requests for servlets
- Manages servlet life cycle for underlying class (e.g. init() and destroy())
- Container:
    - Can have wrapper implementations as child-container
- Implementing classes:
    - StandardWrapper (org.apache.catalina.core)

Container (org.apache.catalina):

- Interface
- Executes requests and creates/returns a response. Alternative: support pipeline of valves for processing
- Possible containers:
    - Engine: Catalina servlet engine, can have hosts or contexts as child-containers
    - Host: Virtual host, can have context as child-containers
    - Context: SingleServlet context, can have Wrapper as child-containers
    - Wrapper: Individual servlet definition, can have wrapper as child-containers
- Implementing classes:
    - ContainerBase (org.apache.catalina.core)
        - "Standard"-classes extend this class to get common functionalities

Realm (org.apache.catalina):

- Read-only facade for security realm

- Authenticate users and their security roles
- Attached at any container level (mostly context or higher)

Pipeline (org.apache.catalina):

- Collection of valves to be executed in order when invoke() is called
- A valve must process a request and create a proper responses
- One pipeline for each Container

Valve (org.apache.catalina):

- Request-processing component associated with its container

Lifecycle (org.apache.catalina):

- Interface for component lifecycle methods
- Not mandatory for Catalina components
- Provides consistent mechanism to start and shutdown component