# Rest Protocol

Iiot Application Protocol Overview of Representational State Transfer (REST) and HTTP,

HTTP is by far the most used application-layer protocol.

HTTP is extendable. ( you can add your own headers, method, not implement some method or the header logic )

REST defines a set of rules and principles that all the elements of the architecture must conform to in order to build web application that scale well, in terms of

* Scalability
* -robustness

Is based on the concept of resource

* Builds on clients (initiate the interaction) and servers (origin, host resource)

Intermediaries act as clients and servers at the same time (reverse proxies appear as origin server to a client, while forward proxies known to client are exit points for a request)

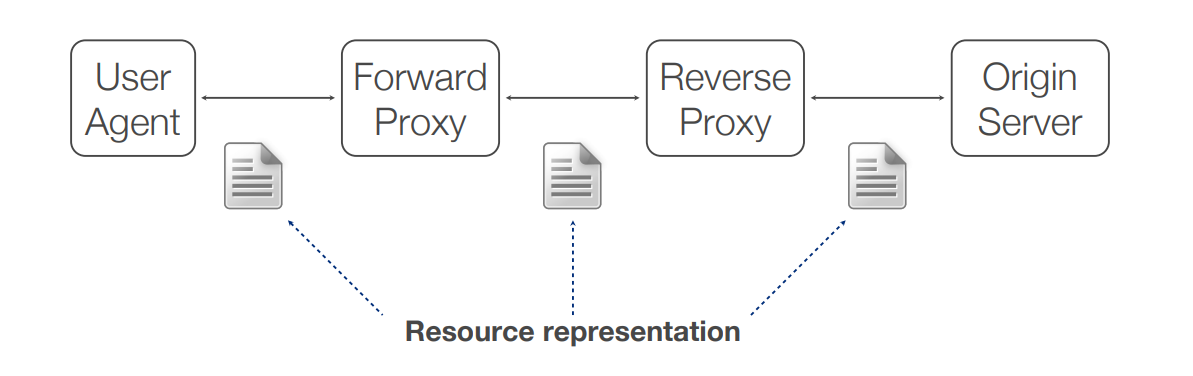
Uniform interfaces all connectors must conform in

~ identification of resources

~ Manipulation of resources through representations

~ self descriptive message

~ hypermedia as the engine of application state



Remember is not the resource that is sent, but his representation

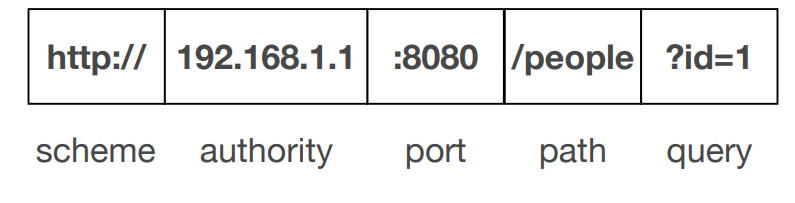
**Resource identifiers**

Uniform Resource Identifiers (**URI**) serve this need

A resource can be mapped to multiple URIs, but a URI points exactly to one resource.

* URN Specifies the name of the resource
* URL specifies how to locate the resource

URI scheme examples



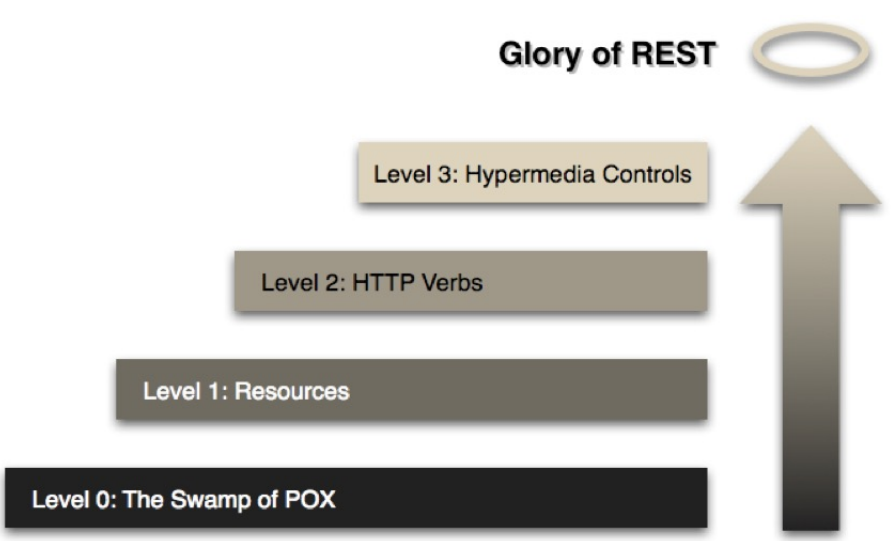
Rest is **stateless**

Requests must be stateless as well.

Requests must contains all the information to understand the request so that servers can process it without context.

Each message should be self descriptive

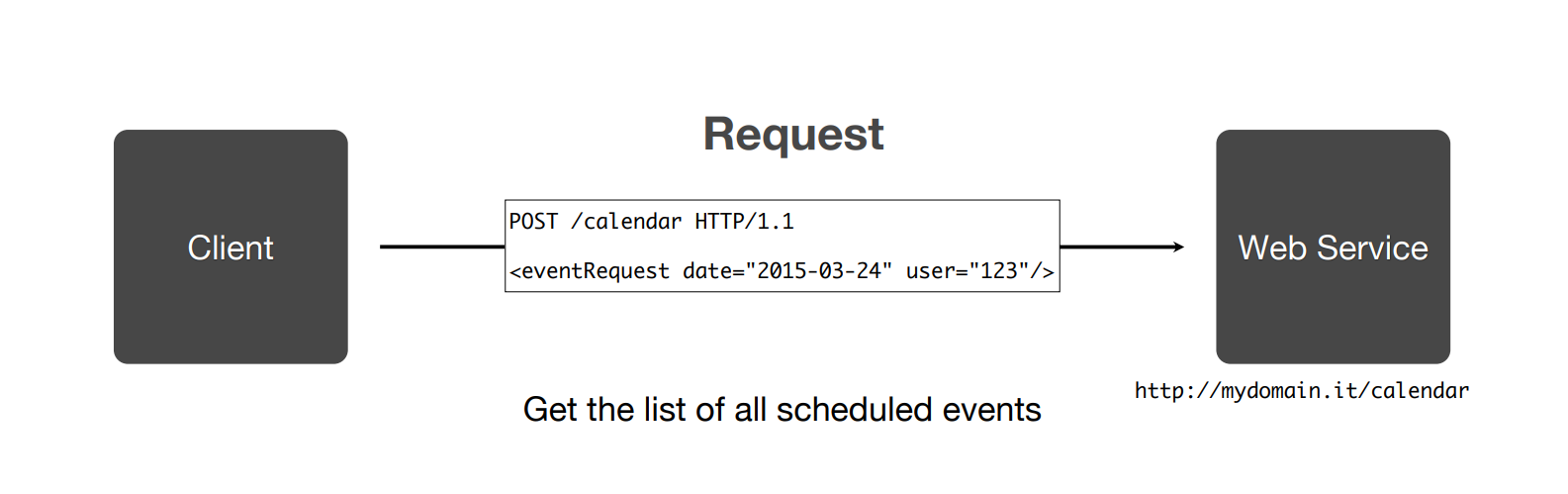
No concept of session



**­­­­Level 0 The Swamp of POX**

Plain old XML -> POX

A single HTTP method is used (POST, the client trigger an action)



It’s just a remote interaction mechanism.

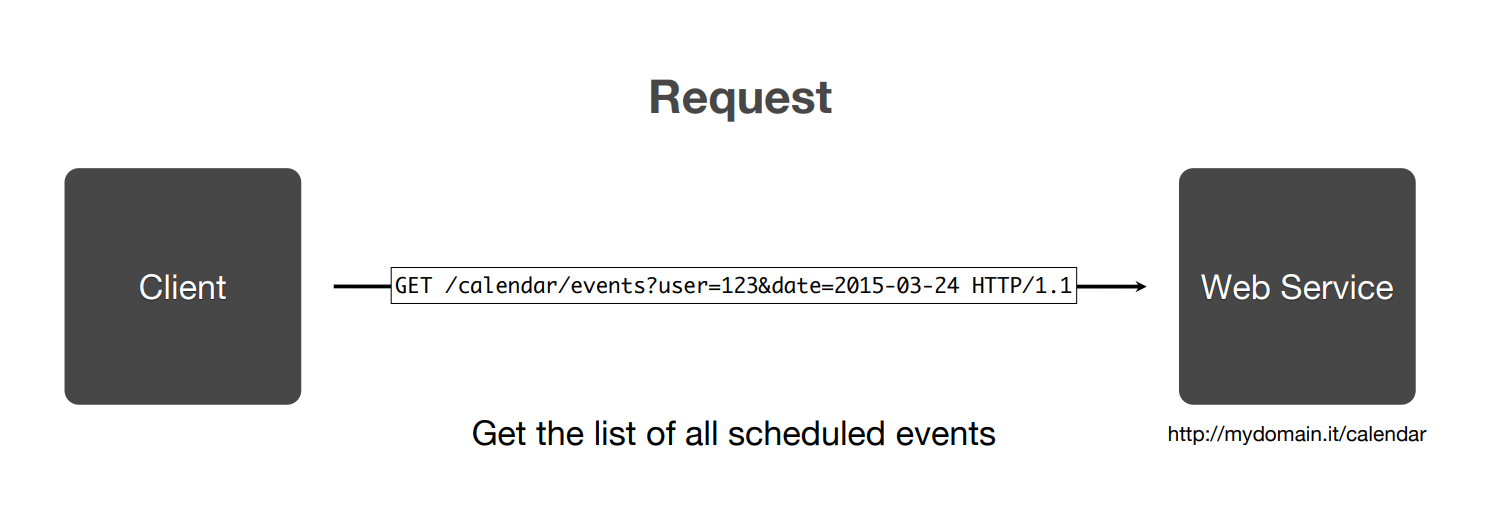
In this case the design is weak and the client must have a very deep knowledge of the web service.

**Level 1 Resource**

Individual resources are addressed and the requested are not from a singular service endpoint

Action name are also mapped in uri instead of xml

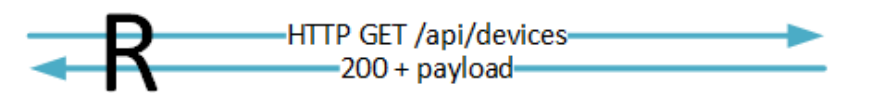
The action is trigger by sending a get or a post to the URI



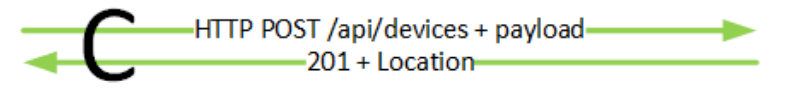
The list of all scheduled events is included in the returned XML.

**Level 2 (HTTP) Verbs**

* GET (safe) is used to retrieve a resource with he given URI



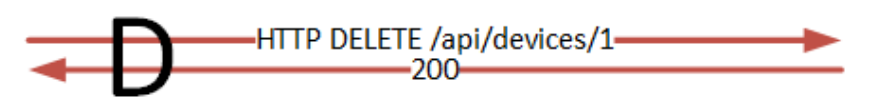
* POST (not safe and not idempotent) is used to create a resource



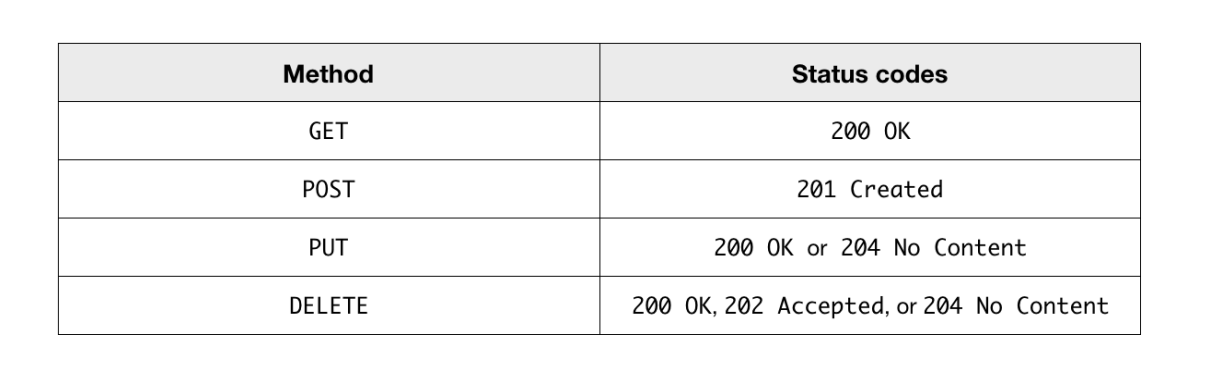
* PUT (not safe and idempotent) is used to update a resource with the given URI



* DELETE (not safe and idempotent) is used to update a resource with the given URI



* **Safe** means that no side-effects occur on the resource
* **Idempotent**  means that multiple request have the same effect of a single request

****

**Level 4 Hypermedia**

Is an extension of hypertext

It Represents a medium information that includes graphic, audio video, text and hyperlinks

Support Hypermedia As the Engine of Application State

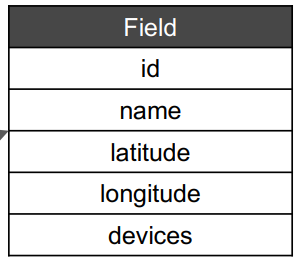
Use hypermedia to link to other resources.

The state of a resource is aggregation of:

* Data: values of information
* Links: representing transitions to possible future state of the current resource

Consumers need to discover and interact with resouces

Go look reedbucks example

**Data Model**

Representations of the structure and attributes of the data entities within an application.

**Data Transfer Objects**

Is the encapsulate subset of the data from the data models.

Example (Json) {“uuid”:”10001”, “name”: “TestBuilding”,”latitude”:48.12231,”logintude” :10.45792}