

RESTful design in IoT

Chapter Contents:-

- What is REST
- REST methods
- REST Architecture
- Uniform Resource Identifiers (URIs)

REST

- Representational State Transfer (REST) is an architectural style consisting of a set of guidelines and best practices for developing distributed hypermedia systems.
- When REST principles are applied to the design of a system, the design is called RESTful .
- Since RESTful APIs are simple and lightweight, they are good fit for IoT applications.
- Protocols used with RESTful systems are HTTP (RFC 7230) and CoAP (RFC 7252)

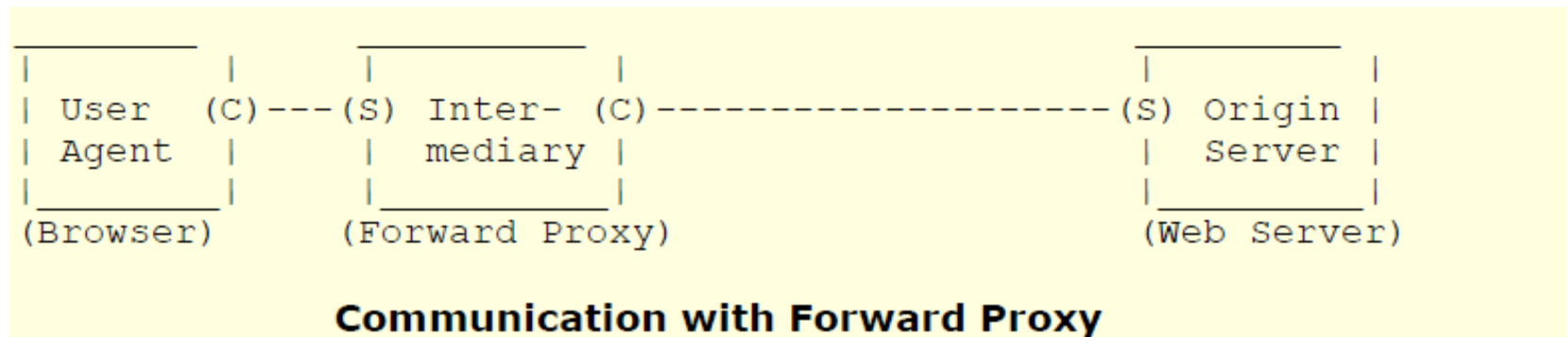
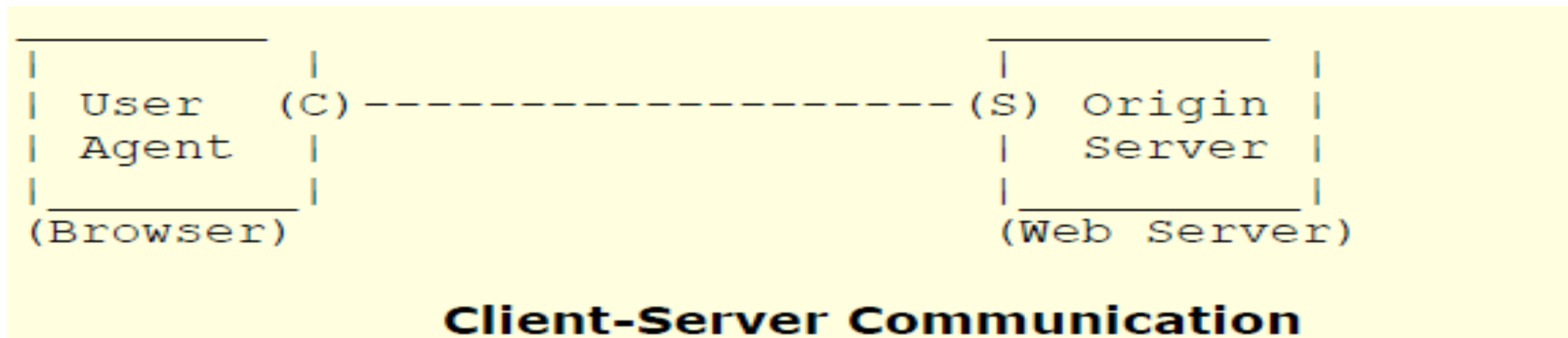
REST Contd.

- The key abstraction in RESTful webservice is a resource, and not the service itself.
- Webservices acts upon resources.
- A resource could be sensor, actuator or a control system.
- A resource is uniquely indentified by its URI
- A webservice can be called RESTful if it conforms to architectural constraints like Client-Server, Stateless, Cacheable, layered system and uniform interface.

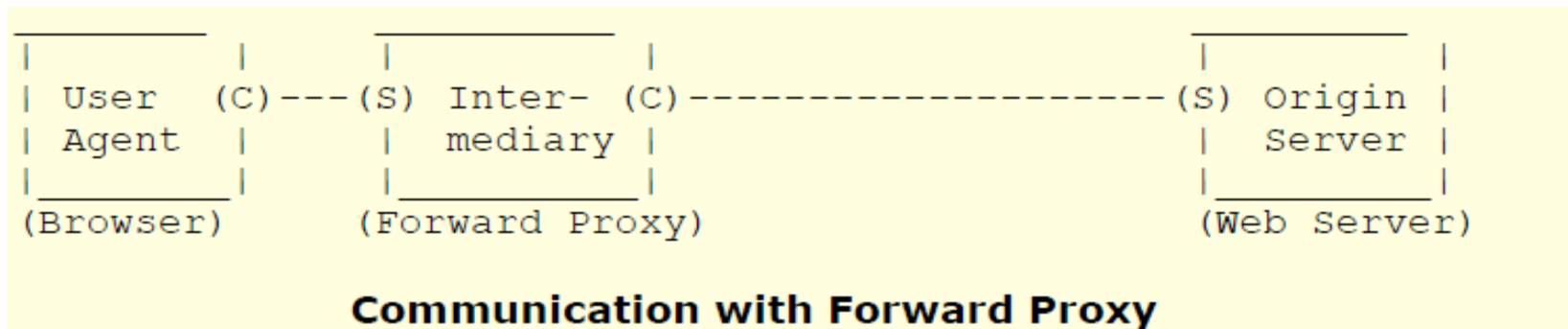
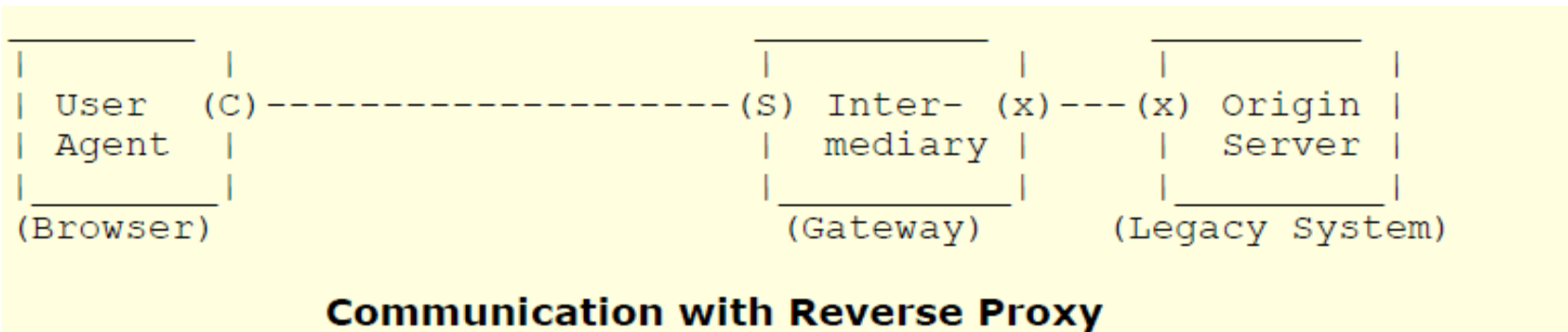
REST Methods

- REST design principle uses the following methods for performing CRUD operations
 - POST – Create a resource
 - GET – Read a resource
 - PUT – Update a resource
 - DELETE – Delete a resource

REST Architecture



REST Architecture Contd.



Uniform Resource Identifiers (URIs)

- An important part of RESTful API design is to model the system as a set of resources whose state can be retrieved and/or modified and where resources can be potentially also created and/or deleted

```
foo://example.com:8042/over/there?name=ferret#nose
  \_/_/      \_____/_/      \_____/_/      \_____/_/
    |         |               |               |         |
scheme authority  path      query      fragment
```

An example of URI Structure