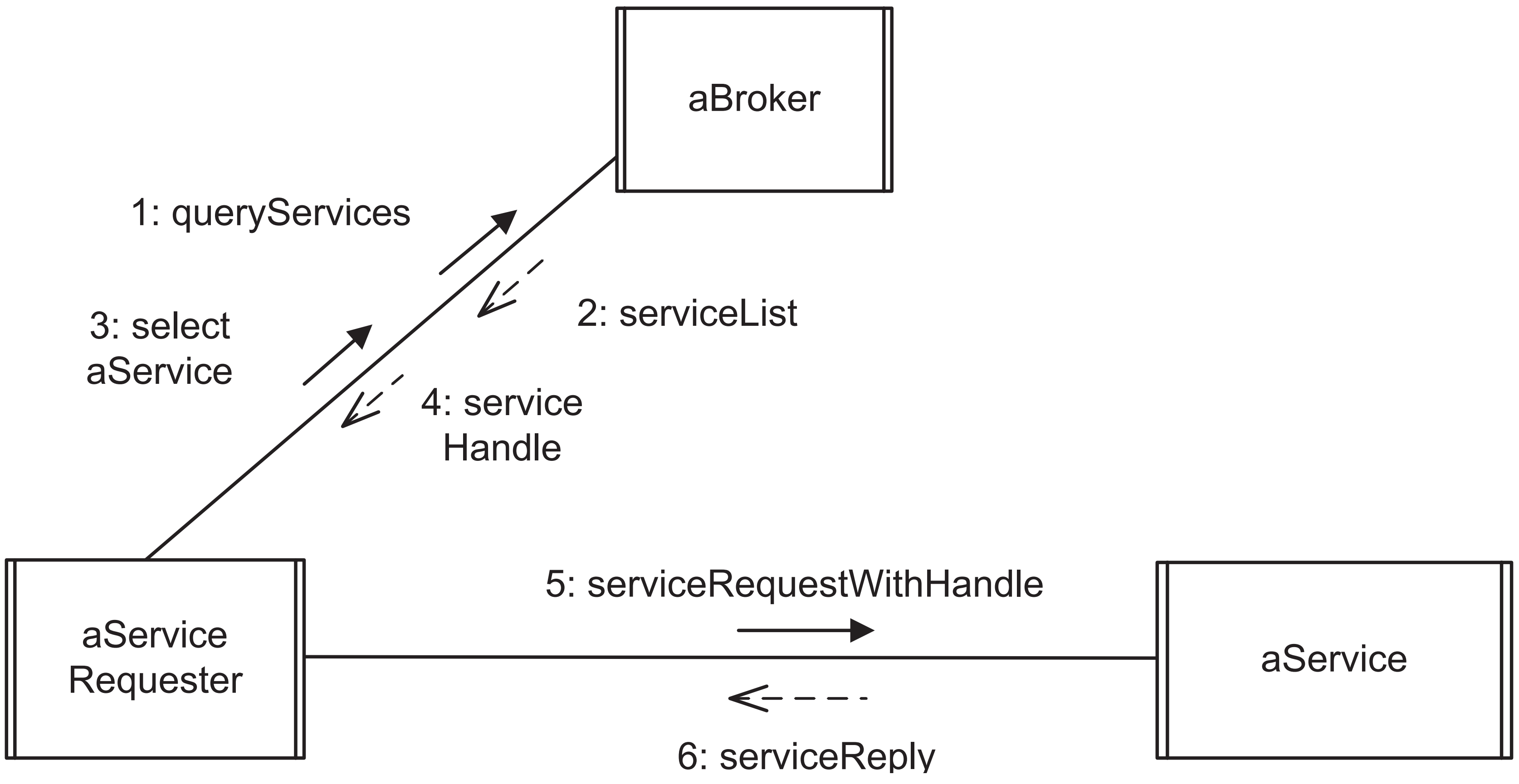
Service Discovery Pattern (yellow pages)

- In *white pages brokering* the client knows the service required but not the location
- A different brokering pattern is *yellow pages brokering*, analogous to the yellow pages of the telephone directory, in which the client knows the type of service required but not the specific service
- Also known as the **Service Discovery** pattern because it allows the client to discover new services:
    1. The client sends a query request to the broker, requesting all services of a given type
    2. The broker responds with a list of all services that match the client's request
    3. The client, possibly after consultation with the user, selects a specific service
    4. The broker returns the service handle, which the client uses for communicating directly with the service

Service Discovery Pattern (yellow pages)



Technology Support for SOA

- Although SOAs are conceptually platform-independent, they are currently provided very successfully on **Web Services** technology platforms
- A *web service* is a service that is accessed using standard Internet and XML-based protocols

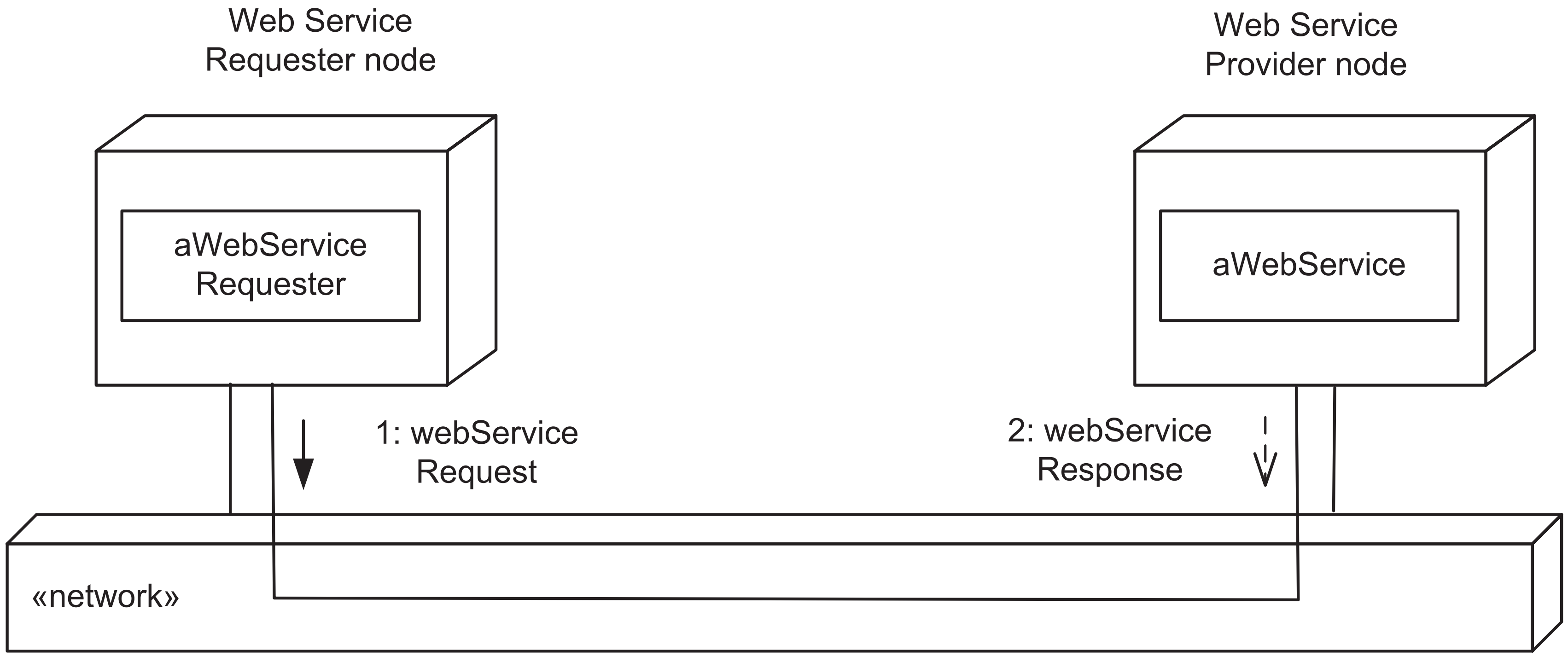
Web Service Protocols

- Application clients and services need to have a communication protocol for inter-component communication
- Extensible Markup Language (XML) is a technology that allows different systems to interoperate through exchange of data and text
- The **Simple Object Access Protocol (SOAP)**, which is a lightweight protocol developed by the World Wide Web Consortium (W3C), builds on XML and HTTP to permit exchange of information in a distributed environment
- SOAP defines a unified approach for sending XML-encoded data and consists of three parts:
  - an envelope that defines a framework for describing what is in a message and how to process it
  - a set of encoding rules for expressing instances of application-defined data types, and
  - a convention for representing remote procedure calls and responses

Web Services

- Applications provide services for clients
- One example of application services is **Web services,** which use the World Wide Web for application-to-application communication
- From a software perspective, Web services are the application programming interfaces (APIs) that provide a standard means of communication among different software applications on the World Wide Web
- From a business application perspective, a Web service is business functionality provided by a company in the form of an explicit service over the Internet for other companies or programs to use
- A Web service is provided by a service provider and may be composed of other services to form new services and applications.

Web Service example



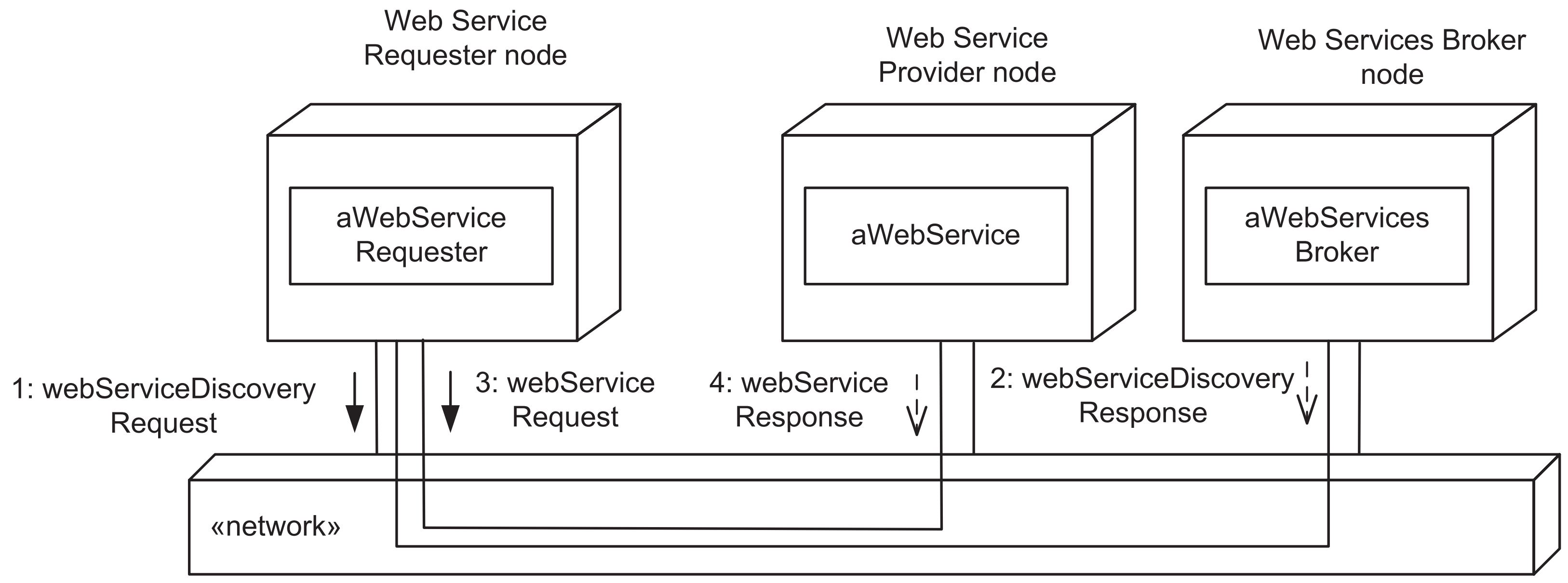
Registration Services

- A registration service is provided for services to make their services available to clients
- Services register their services with a registration service – a process referred to as *publishing* or *registering* the service
- Most brokers, such as CORBA and Web service brokers, provide a registration service
- For Web services, a **service registry** is provided to allow services to be published and located via the World Wide Web.
- Service providers register their services together with service descriptions in a service registry
- Clients searching for a service can look up the service registry to find a suitable service
- The **Web Services Description Language** (**WSDL**) is an XML-based language used to describe what a service does, where it resides, and how to invoke it

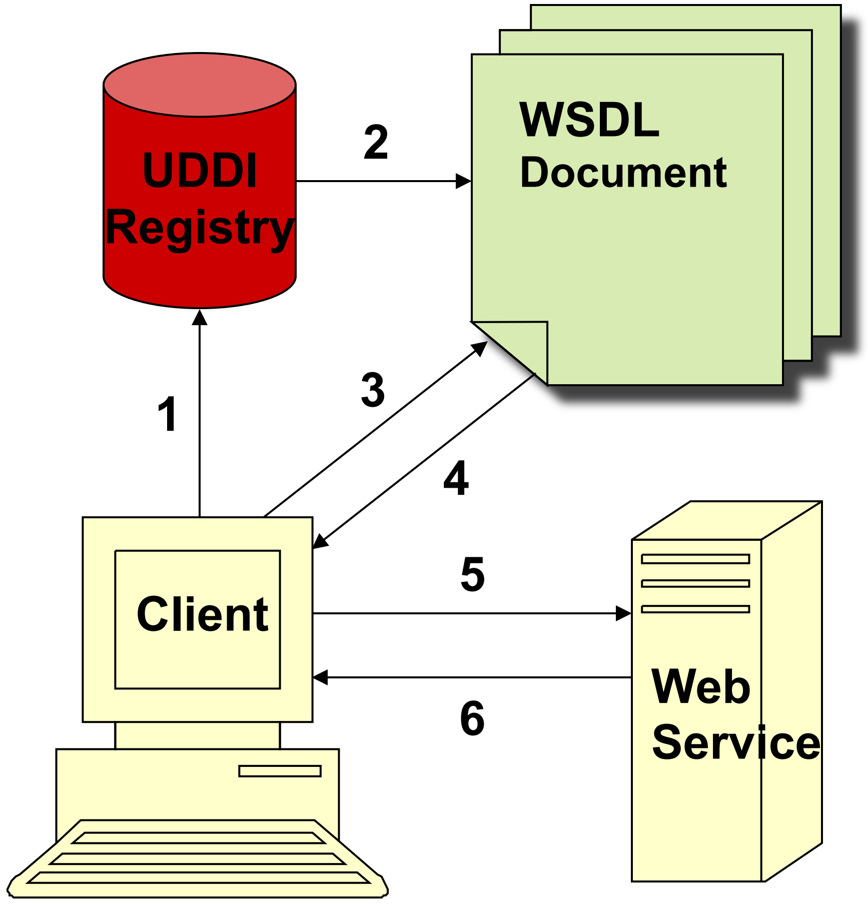
Brokering and Discovery Services

- In a distributed environment, an **object broker** is an intermediary in interactions between clients and services
- An example of brokering technology is a Web services broker
- Information about a Web service can be defined by the **Universal Description, Discovery, and Integration (UDDI)** framework for Web services integration
- A UDDI specification consists of several related documents and an XML schema that defines a SOAP-based protocol for registering and discovering Web services
- A Web services broker can use the UDDI framework to provide a mechanism for clients to dynamically find services on the Web

Web Service Broker Example

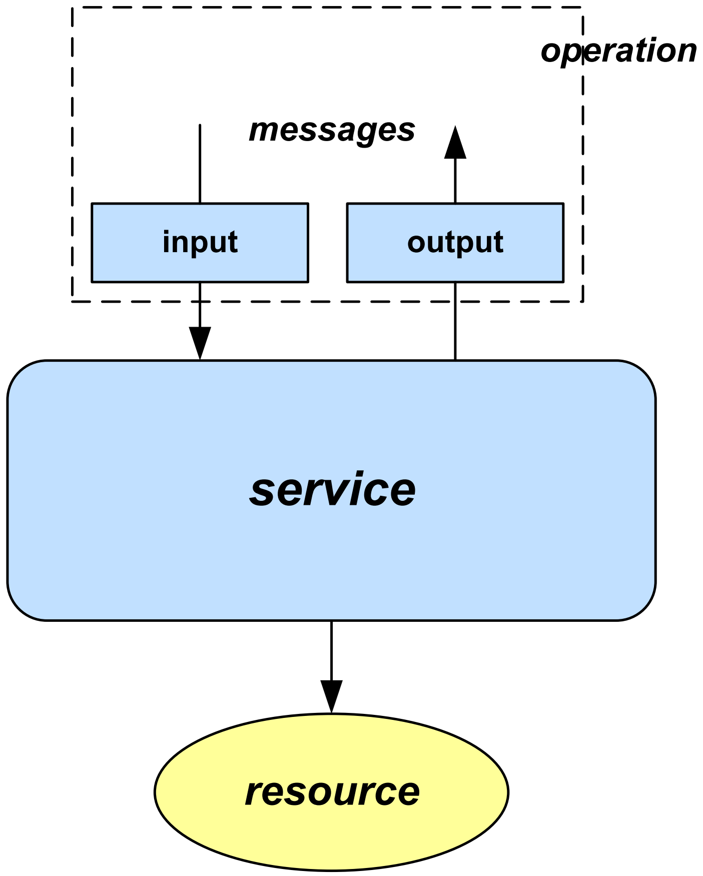


Web Service Protocols and Standards

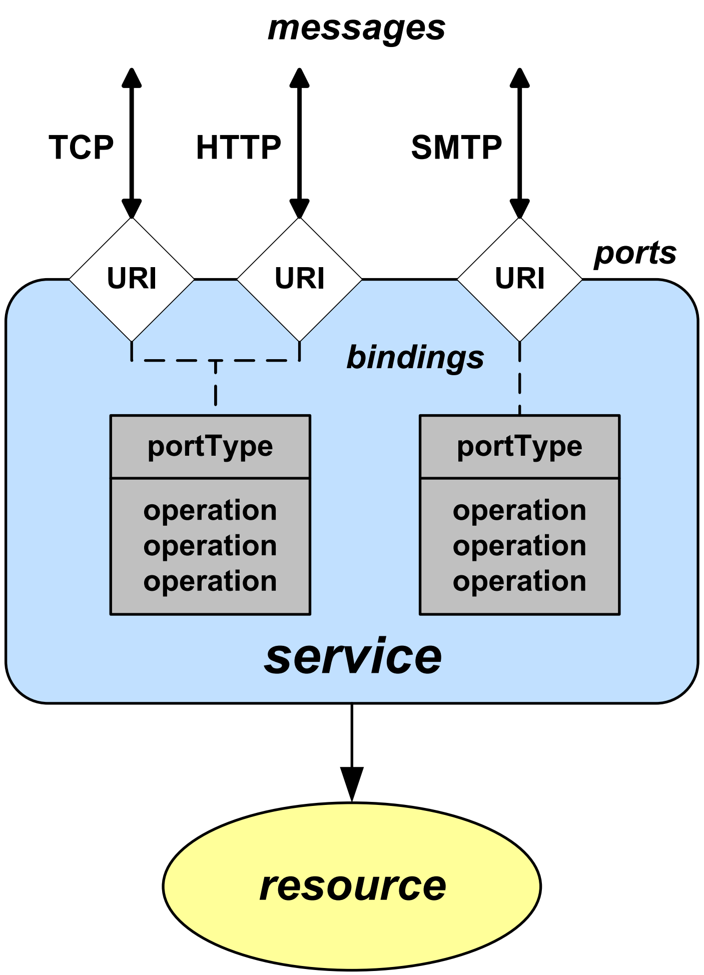


1. Client queries UUDI registry to locate service
2. Registry refers client to WSDL document
3. Client accesses WSDL document
4. WSDL provides data to interact with web service
5. Client sends SOAP-message request
6. Web service returns SOAP-message response

WSDL



Operations



Ports and Bindings

REST

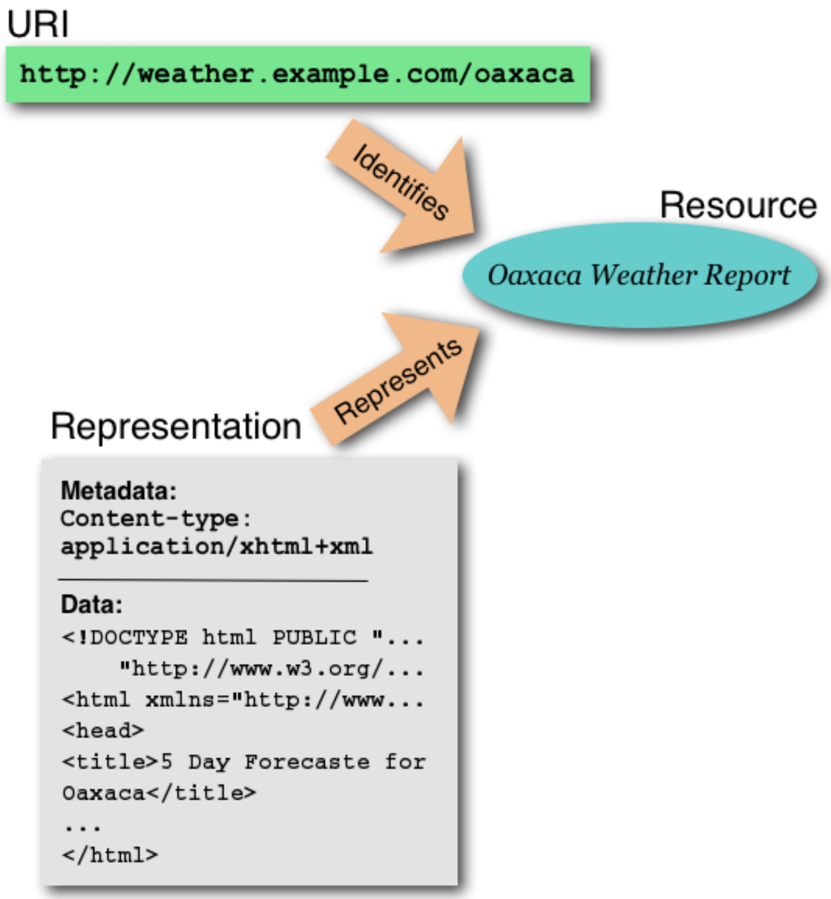
- REST stands for *Representational State Transfer*
- REST is a term coined by Roy T. Fielding to describe an architecture style of networked systems
- RESTful API
  - A resource-based API that uses the HTTP protocol

REST-based network characteristics

- *Client-Server*: a pull-based interaction style
- *Stateless*: the client-server communication is constrained by no client context being stored on the server
- *Cache*: clients and intermediaries can cache responses
- *Uniform interface*: all resources are accessed with a generic interface (e.g., HTTP GET, POST, PUT, DELETE), thus simplifying and decoupling the architecture
- *Named resources:* the system is comprised of resources which are named using a URL (or URI)
- *Interconnected resource representations:* the *representations* of the resources are interconnected using URLs, thereby enabling a client to progress from one state to another

Resources

- Resources
  - every distinguishable entity is a resource.
  - a resource may be a Web site, an HTML page, an XML document, a Web service, a physical device, etc.
- URLs Identify Resources
  - Resources are is uniquely identified by a URL (Axiom 0 of Tim Berners-Lee Web Design)



```
Metadata:
Content-type:
application/xhtml+xml

Data:
<!DOCTYPE html PUBLIC "...
    "http://www.w3.org/...
<html xmlns="http://www...
<head>
<title>5 Day Forecaste for
Oaxaca</title>
...
</html>
```

RESTful API

- The RESTful API uses the available HTTP verbs to perform CRUD operations based on the “context”:
  – *Collection*: A set of items (e.g.: /users)
  – *Item*: A specific item in a collection (e.g.: /users/{id})

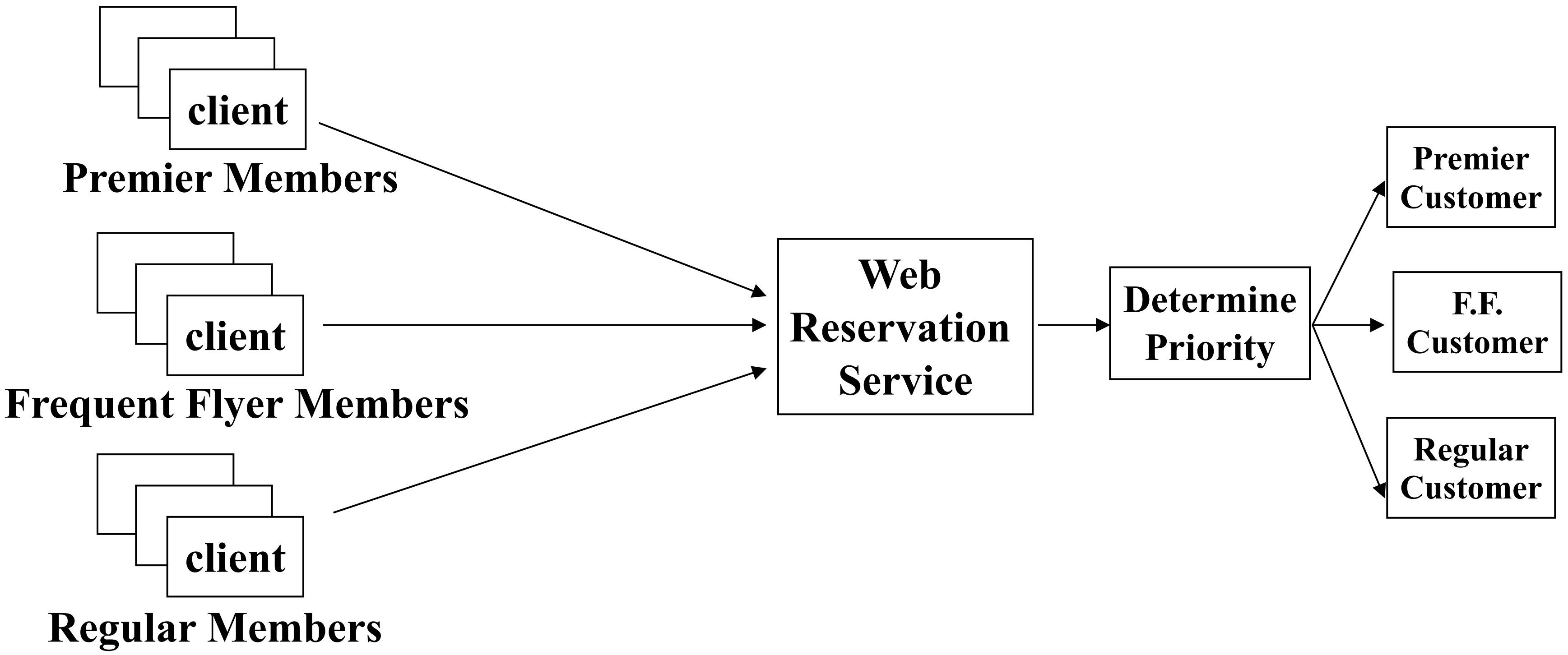
VERB	Collection	Item
POST	Create a new item.	Not used
GET	Get list of elements.	Get the selected item.
PUT	Not used	Update the selected item.
DELETE	Not used	Delete the selected item.

Conventional vs. REST-based design

- Example scenario
  - an airline wants to provide a Web reservation service for customers to make flight reservations through the Web.
  - the airline wants to ensure that its premier members get immediate service, its frequent flyer members get expedited service, all others get regular service.
- Two main approaches to design and implement the Web reservation service
  - Single URL approach: based on conventional web service design
  - Multiple URLs approach: exploits REST-based design

Single URL approach

- The Web service is responsible for examining incoming client requests to determine their priority and process them accordingly

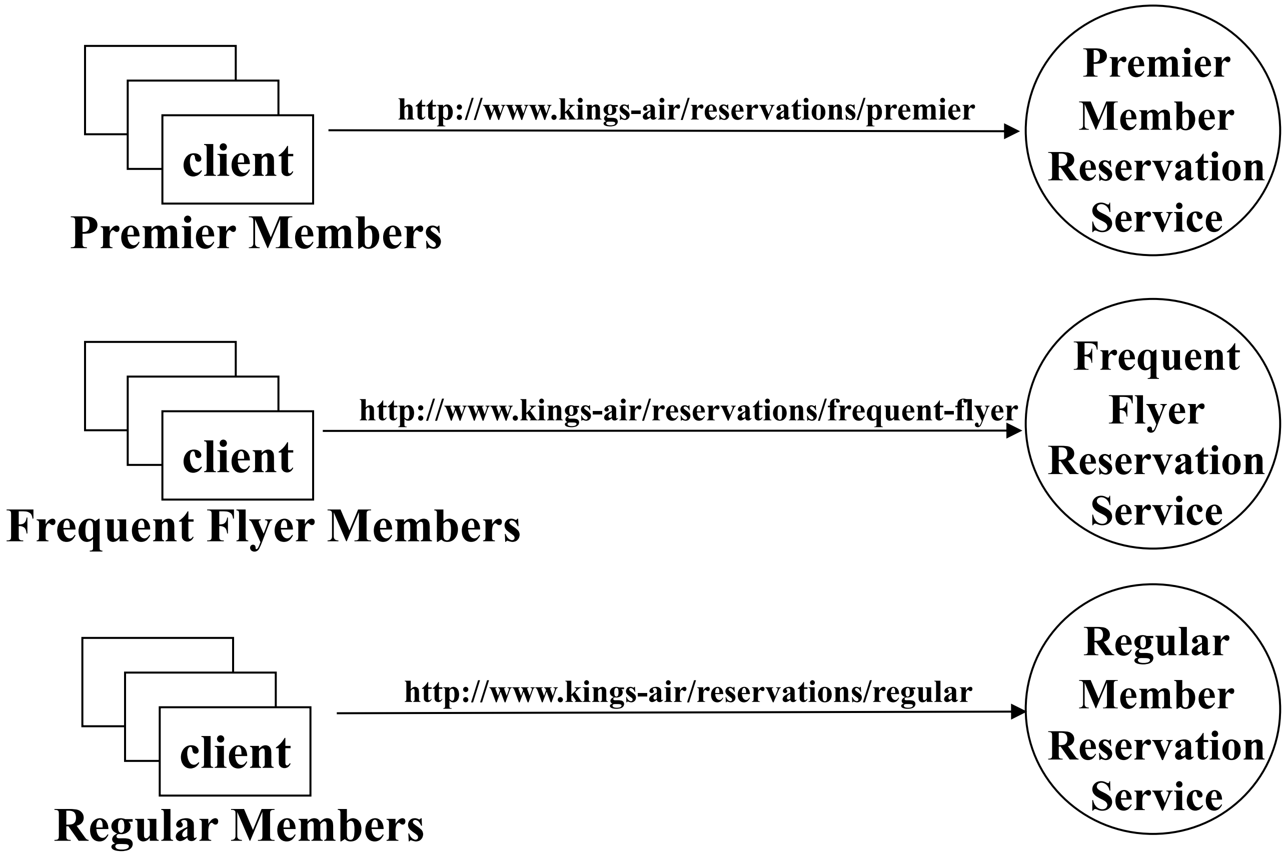


Single URL approach disadvantages

- Clients must learn the rule for expressing priorities, and the Web service application must be written to understand the rule
- Based upon the incorrect assumption that a URL is "expensive" and that their use must be rationed
- The Web service is a central point of failure and a bottleneck
  - Load balancing is a challenge
- It violates Axiom 0 of Tim Berners-Lee Web Design

Multiple URLs approach

- One URL for premier members, a different URL for frequent flyers, and still another for regular customers


client
Premier Members
http://www.kings-air/reservations/premier
Premier Member Reservation Service
client
Frequent Flyer Members
http://www.kings-air/reservations/frequent-flyer
Frequent Flyer Reservation Service
client
Regular Members
http://www.kings-air/reservations/regular
Regular Member Reservation Service

Multiple URLs approach advantages

- It's easy to understand what each service does simply by examining the URL
- There is no need to introduce rules
  - Priorities are elevated to the level of a URL. "What you see is what you get"
- It's easy to implement high priority
  - simply assign a fast machine at the premier member URL.
- There is no bottleneck and no central point of failure
- Consistent with Axiom 0