1. web service :
a. SOAP
b. REST :

Open API Spec. : it defines the std. and language agnostic interface f
RESTful APIs , allowing humans and computers to discover and
understand service capabilities without requiring access to the source c
or any additional doc.

1. any awareness regarding coding platform
2. when we will write this specs we will get the std template with respec
multiple platforms.
3. we can write codes and we can verify the expectations using this op
api. details.

benefits :
1. Standardization : provide the common format for defining the apis
leading to consistency in api design and docs.
2. Automation : tools can auto gen doc. stubs (server / client) manua
efforts and potentials errors.
3. API life cycle management :
a. planning :

- **Requirement Gathering:** Identify the business requirements, use
cases, and target audience for the API.
- **Defining Objectives:** Set clear goals for what the API should achie
such as data sharing, service access, or partner integration.
- **Design Considerations:** Consider performance, scalability, and sec
needs.
- **Technology Stack:** Select the technology, framework, and platform
development (e.g., REST, GraphQL, SOAP).

b. design :
i. api spec: end points, resources, methods req/res formats . Open AF
Swagger.
ii. security Design : (public/private (TOKEN /JWT/ OAUTH) authenticat
policy(RBAC / ABAC -> (Attribute based access control / PBAC (specific

or

ode

t to

en

,

al

e

ve,

urity

for

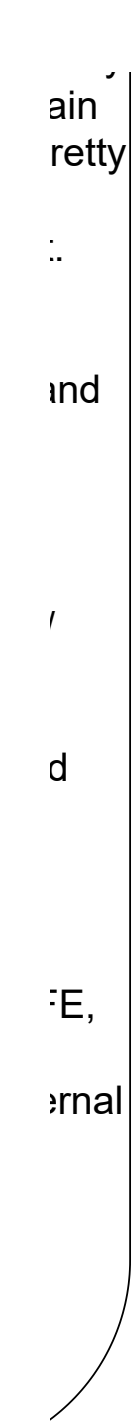
I/

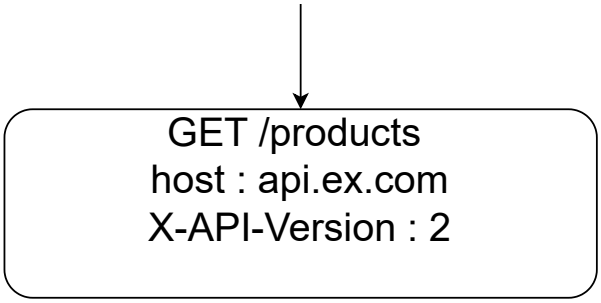
ion
cally

based on the region can access the specific resources during the certa
hrs limitations.) => user attributes / resource attributes / env attributes pı
dynamic in nature
ex : a policy can allow access to the resources based on user's dept
location or time of req.
iii. Versioning Strategy : header versioning :
iv. Error Handling and status codes : define consistent error handling a
status codes for diff api responses.
. visibility of the end point).
c. development
d. testing :
e. deployment Automate the API dev. using tools Jenkins , gitlab CI/
GitHub.
Env : dev prod staging .
f. management and monitoring. versioning , monitoring (unauthorise
access/ measure api key usage)
g. Maintenance
h. retirement

3. Ease of communication : simplifies the comm between teams(BE, F
Testing, DevOps) , enabling the smooth collaborations.
4. Adoption : Industry Stds, apis are more like to be adopted by the exte
developers , leading to have better integration and usage.

GET /products
host : api.ex.com
X-API-Version : 1

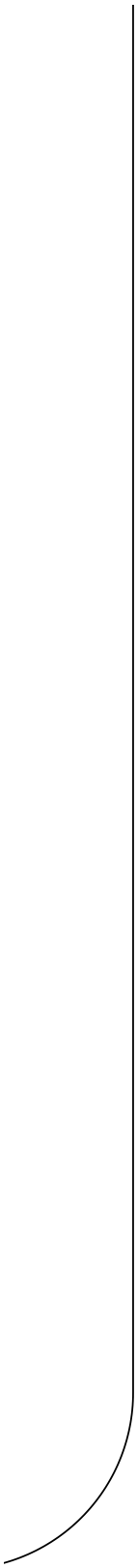

ain
retty
..
and
d
E,
rnal


GET /products
host : api.ex.com
X-API-Version : 2

```
{
  "policyId": "FinanceDocumentAccessPolicy",
"description": "Controls access to confidential financial docum
  "rules": [
    {
      "effect": "Allow",
      "conditions": [
        {
          "subject": {
            "department": "Finance"
          },
          "resource": {
            "type": "document",
            "classification": "confidential"
          },
          "environment": {
            "ipAddress": "inSubnet('192.168.1.0/24')",
            "timeOfAccess": "between('09:00', '18:00')"
          }
        }
      ]
    }
```

nents.",

```
},
{
"effect": "Deny",
"conditions": [
{
"subject": {
"department": "Finance"
},
"environment": {
"ipAddress": "not inSubnet('192.168.1.0/24')"
}
}
]
}
]
}
```



2: Core Definitions and Components (2 Hours)

- 2.1 Definitions
  - o OpenAPI Document
  - o Path Templating
  - o Media Types
  - o HTTP Status Codes
- 2.2 Specification Components
  - o Versions
  - o Format
  - o Document Structure
  - o Data Types
  - o Rich Text Formatting

3: Schema and Object Structures