SNJB’s KBJ College of Engineering

Chandwad–423101 (Nashik)

Department
of
Information Technology

Subject : Internet of Things (IOT) of BE 2015 Pattern

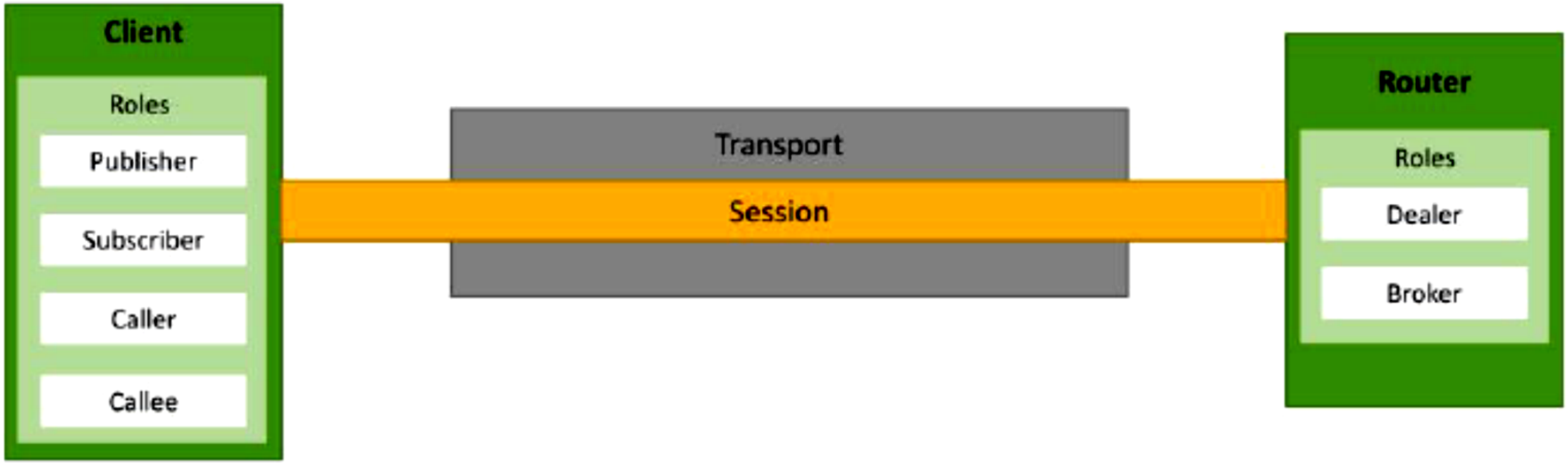
Unit 6
IoT PHYSICAL SERVERS AND CLOUD OFFEREINGS

Introduction to Cloud Storage Models and Communication API's

- Cloud computing is a transformative computing paradigm that involve delivering applications and services over the Internet
- Here we will see how to use cloud computing for IOT
- We will see WAMP, Xively's Platform-as-a-Service (PaaS)
- Also, AWS & their applications for IOT

WAMP-AutoBahn for IoT

- Web Application Messaging Protocol
- Sub-protocol of Web socket which provides
  - Publish-Subscribe and
  - Remote Procedure Call (RPC) messaging patterns



WAMP-AutoBahn for IoT

- Transport: Channel that connects two peers
- Session: Conversation between two peers that runs over a transport
- Client: Peers that can have one or more roles
- In publish-subscribe model client can be:
  - Publisher: Publisher publishes events (including payload) to the topic maintained by the Broker
  - Subscriber: Subscriber subscribes to the topics and receives the events including the payload

WAMP-AutoBahn for IoT

- In RPC model client can be:
  - Caller: Issues calls to the remote procedures along with call arguments
  - Callee: Executes the procedures to which the calls are issued by the caller and returns the results back to the caller
- Router: Routers are peers that perform generic call and event routing

WAMP-AutoBahn for IoT

- In publish-subscribe model Router has role of Broker, acts as router and routes messages published to topic to all subscribers subscribed to the topic
- In RPC model Router has the role of a Broker:
  - Dealer: acts as router, routes RPC calls from Caller to Callee & routes results from Callee to Caller
- Application Code: Application code runs on the Clients (Publisher, Subscriber, Callee or Caller)

Python Web Application Framework

- Python Web Application Framework – Django
- Its an open source web application framework for developing web applications in Python
- A web application framework in general is a collection of solutions, packages and best practices that allows development of web applications and dynamic websites

Python Web Application Framework

- Django is based on Model-Template-View architecture and provides separation of data model from business rules and user interface
- It provides unified API to a database backend
- Thus web applications built with Django can work with different databases without requiring any code changes

Python Web Application Framework

- With this flexibility in web application design combined with the powerful capabilities of the Python language and the Python ecosystem, Django is best suited for cloud applications
- Django consists of an object-relational mapper, a web templating system and a regular-expression based URL dispatcher

Django Architecture

Model

- Acts as definition of stored data and handles interactions with database
- In a web application, data can be stored in a relational database, non-relational database, an XML file, etc.
- Django model is Python class that outlines variables and methods for particular type of data

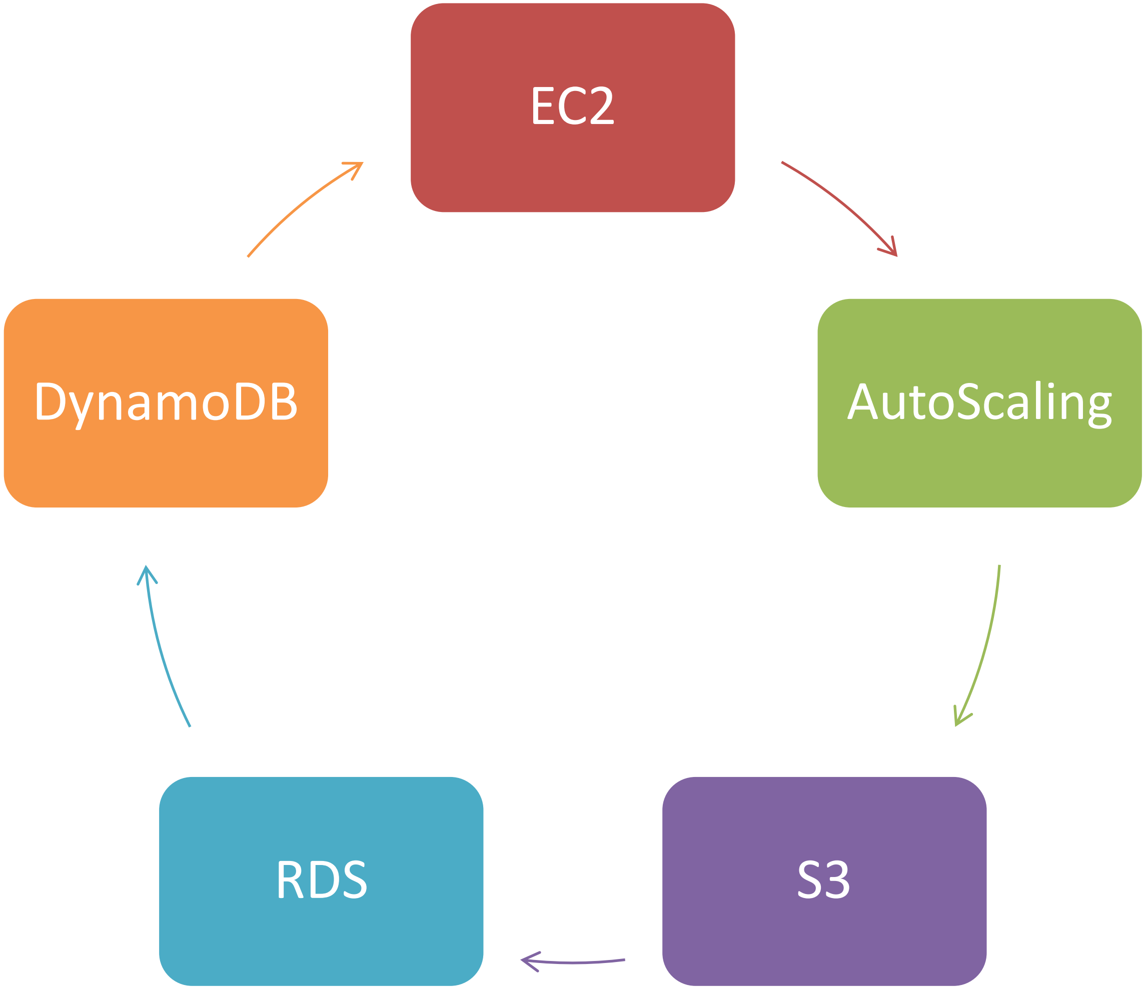
Template

- In a typical Django web application, the template is simply an HTML page with a few extra placeholders
- Django’s template language can be used to create various forms of text files (XML, email, CSS, JavaScript, CSV, etc.)

View

- The view ties the model to the template
- The view is where you write the code that actually generates the web pages

AMAZON Web Services for IoT


EC2

AutoScaling

S3

RDS

DynamoDB

Amazon EC2

- An IaaS provided by Amazon
- EC2 delivers scalable, pay-as-you-go compute capacity in the cloud
- Web service that provides computing capacity in the form of virtual machine
- EC2 can be used for several purposes for IOT systems

Amazon AutoScaling

- Allows automatically scaling EC2 capacity up or down according to user condition
- Users can increase number of EC2 instances
- AutoScaling can be used for auto scaling IOT applications and IOT platforms deployed as Amazon EC2

Amazon S3

- Online cloud based data storage infrastructure for storing and retrieving large amount of data
- Offers reliable, scalable, fast, fully redundant and affordable storage infrastructure
- Serve as raw datastore for IOT systems for storing raw data such as sensor data, log data, image, audio, video, etc.

Amazon RDS

- Web service that allows to create instances of MySQL, Oracle or MS SQL Server in cloud
- Developers can easily set up, operate and scale a relational database in cloud
- Serve as a scalable datastore for IOT systems
- With RDS, IOT system developers can store any amount of data in scalable relational databases

Amazon DynamoDB

- Fully-managed, scalable, high performance No-SQL database service
- Serve as scalable datastore for IOT systems
- With DynamoDB, IOT system developers can store any amount of data and serve any level of requests for the data

SkyNet IoT Messaging Platform

- Open source instant messaging platform
- SkyNet API supports both HTTP REST and real-time WebSockets
- SkyNet allows you to register devices on the network
- A device can be anything including sensors, smart home devices, cloud resources, drones
- Each device has an UUID & secret token

IoT Case Studies

- Home Automation
- Cities
- Environment