 **COMSATS UNIVERSITY ISLAMABAD, ABBOTABAD**

**Assignment: SDA**

**Task: "TASK"**

**Submitted By:** **JALAL KHAN**

**HAIDER REHMAN**

**REG NO: FA22-BSE-093**

**FA22-BSE-064**

**Department of Software Engineering**

**BSE4A**

**Netflix: Version History and Architecture**

Netflix doesn't use traditional version numbers (like 3.3.2) since it updates its system continuously. But if we imagine version numbers for its major changes, here's how it could look:

**Netflix's Hypothetical Versions**

**1. Initial Launch (1997 – DVD Rental Service) Version 1.0 :**

* **Architecture Used**:
  + Initially based on a **monolithic architecture**. Netflix operated as an e-commerce platform for DVD rentals using a traditional web stack. The backend likely relied on a simple relational database (e.g., Oracle DB) to manage inventory, users, and orders.
* **Version Features**:
  + Users could browse DVDs on the website, order them online, and receive them via mail.
* **Need**:
  + Addressed the inconvenience of traditional video rental stores, offering a home delivery solution without late fees.

**2. Transition to Online Streaming (2007) Version 2.0 :**

* **Architecture Used**:
  + Adopted a **monolithic architecture** but started to face scalability issues due to growing demand.
  + Hosted on **data centers** managed by Netflix itself.
* **Version Features**:
  + Users could stream select movies and TV shows directly on their computers via the internet.
  + Introduced **recommendation algorithms** to improve user experience.
* **Why This Version?**
  + The rise of broadband internet created an opportunity to shift from physical media to digital content delivery.

**3. Move to Cloud (2010 – International Expansion) Version 3.0:**

* **Architecture Used**:
  + Transitioned from monolithic to a **service-oriented architecture (SOA)** and moved to **Amazon Web Services (AWS)**.
  + Netflix began using the **cloud** to improve scalability, reliability, and reduce dependency on physical data centers.
* **Version Features**:
  + International streaming launched, starting with Canada.
  + Enhanced user interface to handle global audiences.
* **Why This Update?**
  + Rapid growth in users and global demand required scalable and cost-effective solutions.

**4. Shift to Microservices (2012 – Improved Scalability) Version 3.1**

* **Architecture Used**:
  + Migrated to a **microservices architecture** to address issues with scaling a large user base.
  + Used APIs for communication between independent services (e.g., user data, recommendations, playback).
  + Introduced tools like **Chaos Monkey** to ensure system resilience.
* **Version Features**:
  + Personalized recommendations powered by machine learning.
  + Support for multiple devices (smart TVs, smartphones, tablets).
* **Why This Version?**
  + The monolithic approach couldn’t handle the increasing complexity of a global streaming platform.

**5. Introduction of Original Content (2013 – House of Cards) Version 3.2:**

* **Architecture Used**:
  + Continued refining the microservices approach, leveraging big data for decision-making.
  + Used **Apache Cassandra** for handling large-scale data storage and **Apache Kafka** for real-time data streaming.
* **Version Features**:
  + Launched Netflix Originals, starting with *House of Cards*.
  + Improved algorithms to tailor recommendations for new content.
* **Why This Version?**
  + Original content reduced reliance on third-party licensing and attracted more subscribers.

**6. Global Expansion (2016 – Worldwide Streaming) Version 3.3:**

* **Architecture Used**:
  + Full microservices deployment, powered by AWS and advanced content delivery networks (CDNs).
  + Developed its own **Open Connect** system to efficiently deliver content globally.
* **Version Features**:
  + Expanded streaming to 190+ countries.
  + Optimized streaming quality for low-bandwidth regions.
* **Why This Version?**
  + To capture a global audience and compete with regional streaming platforms.

**7. Diversification (2021 – Netflix Games) Version 3.4: `**

* **Architecture Used**:
  + Continued leveraging microservices for scalability and introduced infrastructure to support gaming.
  + Integrated AI-powered recommendation systems for gaming content.
* **Version Features**:
  + Added mobile games to the platform, targeting younger audiences.
* **Why This Version?**
  + To compete with gaming platforms and offer diversified entertainment options.

**Netflix Architecture Timeline**

| **Year** | **Architecture** | **Key Features** |
| --- | --- | --- |
| 1997 | Monolithic | DVD rentals, basic database system. |
| 2007 | Monolithic | Online streaming, basic scalability. |
| 2010 | Service-Oriented Architecture (SOA) | Cloud migration to AWS, international expansion. |
| 2012 | Microservices Architecture | Scalability, resilience (Chaos Monkey), personalized recommendations. |
| 2013 | Advanced Microservices | Original content production, real-time data streaming with Kafka. |
| 2016 | Optimized Microservices | Global reach, Open Connect CDN for efficient content delivery. |
| 2021 | Microservices + Gaming Infrastructure | Introduction of Netflix Games, AI-driven personalization. |

**Summary of Evolution Drivers**

1. **Need for Scalability**: The rapid increase in users made monolithic systems insufficient.
2. **Shift to Streaming**: Consumer behavior shifted from physical rentals to on-demand content.
3. **Competition**: Emergence of Amazon Prime, Hulu, and Disney+ required innovation and diversification.
4. **Globalization**: Expanding to international markets required new solutions like Open Connect CDN.
5. **Technology Advancements**: Cloud computing, AI, and big data drove the evolution of Netflix’s architecture.