



Horizon Payment Switch

Technology Selection Strategy

Document Version: 1.0

Release Date: July 22, 2021

Document Change History				
No.	Doc Version	Release Date	Author	Change Description
1	1.0	2021-07-22	Mahyar Esteki	Create First Version

Contents

1 Abstract.....4

2 Prioritize.....4

3 Risks.....5

4 Factors.....5

5 Tools and Technologies Selection.....6

 5.1 Programming Language.....6

 5.2 Database.....8

 5.3 MQ System.....9

 5.4 In-memory Database and Cache.....10

6 Conclusion.....10

7 References.....11

1 Abstract

Proper knowledge of the infrastructure and architecture of any system will lead to a better understanding of its performance. This information helps developers to reach a common view on the use and operation of the system. The purpose of presenting the architectural document of this project will be to create a common understanding of the structure and operation of the system. In this document you will get acquainted with system architecture, how services work, network architecture and more. This document contains general information and brief descriptions of the system architecture. Therefore, additional technical information will be provided in the relevant documents.

2 Prioritize

Considering the records of banking software produced in Iran, efforts were made to eliminate the weaknesses of such products to be a priority of this project. Weaknesses of most banking software in Iran are:

- High complexity
- Low scalability
- Low flexibility
- Low maintainability

As a result, a set of priorities have been identified for the Horizon project that influence the decision to select the appropriate technology for its development. These priorities are listed in the table below.

No	Title	Priority
1	Maintainability	High
2	Flexibility	High
3	Scalability	High
4	Reliability	High
5	Performance	Medium
6	Security	Medium
7	Easy-to-use for Developers	Medium
8	Easy-to-use in Administration	Low

Table 1: Horizon Project Prioritize

Putting priorities such as performance and security in the "medium" level does not mean that these two issues are less significant in the banking systems. Rather, only in this project it is clear that

the impact of these two issues will be more on architecture and will have less impact on the choice of project development technology.

3 Risks

A set of technical risks can pose a serious challenge to a software project. Some of these risks depend on the technologies selected for system development. The risks of each software project are different from other projects based on its nature. The risks that are effective in choosing the technology of this project are:

- **Open Source and Free Software Project:** Because the project is defined as open source, it is not possible to select proprietary or affiliate tools and technologies.
- **Lack of manpower:** This project is initially defined as a personal activity and does not have a complete development team.
- **Lack of time for development:** According to the previous case, this project is personally defined and leisure time is used for its development.
- **Lack of sufficient budget:** This project is developing without using effective organizational budgets.

The above set of risks shows that the use of proprietary technologies, which complicate the architecture and programming code, does not make any sense in the Horizon project.

4 Factors

Also, very different parameters are considered to select the required tools. Wrongly, in many fintech projects, the priorities revolve more around performance. This is a serious strategic mistake that can pose unimaginable challenges for the long-term development and maintenance of a system. Some of these factors based on common patterns of software engineering are:

- Scalability
- Maintainability
- Reliability
- Trainable
- Expandable
- Malleable
- Simplify

Unfortunately, many software companies have become so involved in the concepts of agile development that they ignore software engineering patterns in initial reviews and technical evaluations. Attempts have been made to avoid this approach in the Horizon project.

5 Tools and Technologies Selection

In this section, general studies have been conducted to select appropriate tools for the development of the Horizon project. The studies will be based on sensitive parameters of the project such as the amount of risks of use, non-exclusivity, cost of use, lack of complexity, ease of use and so on. This study has been performed on all platforms and infrastructures used in Horizon project. The results of these studies can be seen below.

5.1 Programming Language

By default, the first and most influential parameter in choosing programming languages for payment switches is "performance". Therefore, languages such as Java, C and C++ are immediately given priority. But it must be said that this is a very big tactical mistake by the managers of a software project. Because the US space agency (NASA) violated this hypothesis by choosing the Python language in the development of space shuttle operational projects in 1994. [1] The fact was that NASA, with its most advanced computers, did not care about the performance of the programming language at the time. Something that has happened today on the servers of large organizations and even personal computers.

Today, hardware features cover the functional shortcomings of programming languages. Therefore, other parameters can be seriously considered. Many software project managers have found that a successful application is created by balancing most of the major software development factors. Some of these parameters are:

- reliable
- Flexibility
- Maintainability
- Ability to work with big data
- Cross Platform
- Ability to work with or as OpenAPI
- No dependence on monopolies

As a result, linguistics should be selected as the main infrastructure of such projects, which, along with proper performance, have the most possible features and the least complexity. Table 2

clearly shows that the Python programming language is nothing less than other programming languages in terms of features and functionality.

Language	Intended use	Imperative	Object-oriented	Functional	Procedural	Generic	Reflective	Event-driven	Other paradigms	Standardized
C	Application, system, general purpose, low-level operations	Yes			Yes					1989, ANSI C89, ISO C90, ISO C99, ISO C11, ISO C18
C++	Application, system	Yes	Yes	Yes	Yes	Yes				1998, ISO/IEC 2003, ISO/IEC 2011, ISO/IEC 2014, ISO/IEC 2017
C#	Application, RAD, business, client-side, general, server-side, web	Yes	Yes	Yes	Yes	Yes	Yes	Yes	structured, concurrent	2000, ECMA, ISO
Go	Application, web, server-side	Yes			Yes		Yes	Yes	concurrent	<i>De facto</i> standard via Go Language Specification
Java	Application, business, client-side, general, mobile development, server-side, web	Yes	Yes	Yes	Yes	Yes	Yes	Yes	concurrent	<i>De facto</i> standard via Java Language Specification
JavaScript	Client-side, server-side, web	Yes	Yes	Yes	Yes		Yes	Yes	prototype-based	1997, ECMA
Kotlin	Application, mobile development, server-side, client-side, web	Yes	Yes	Yes	Yes	Yes	Yes	Yes		No
Python	Application, general, web, scripting, artificial intelligence, scientific computing	Yes	Yes	Yes	Yes	Yes	Yes	Yes	aspect-oriented	"De facto" standard via Python Enhancement Proposals (PEPs)
Ruby	Application, scripting, web	Yes	Yes	Yes			Yes		aspect-oriented	2011(JIS X 3017), 2012(ISO/IEC 30170)
Rust	Application, server-side, system, web	Yes	Yes	Yes	Yes	Yes	No	Yes	concurrent	No

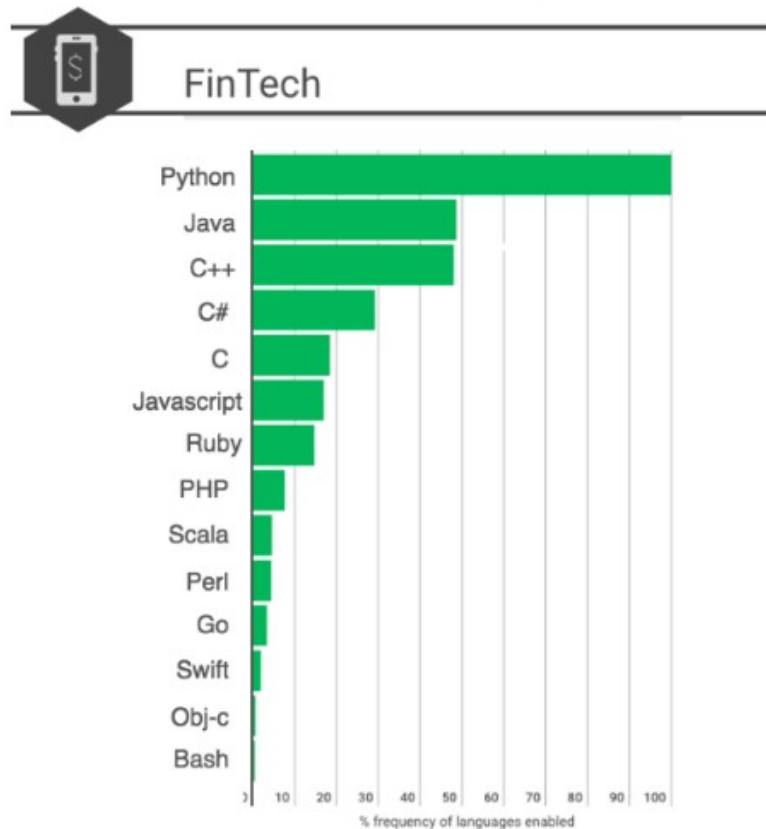
Table 2: Programming Languages Comparison [2]

According to research, a large number of programs produced in the field of Fintech in the United States are based on the Python programming language. [3] The global FinTech market was valued at \$127.66 billion and is expected to reach \$309.98 billion at a CAGR of 24.8%, a report by PRNewsWire suggests. The staggering growth is the result of digital payments and transactions. Some of the already popular Python-based solutions in FinTech are Stripe, Robinhood, Zapa, Square, Paypal, amongst many others. [4]

As Bank of America Merrill Lynch's managing director and head of global risk systems, Singh built the bank's main platform for pricing trades, managing positions and computing risk exposure across all asset classes. Started in 2010 and named Quartz, this project is believed by many observers to mark the first significant arrival of Python in the sector. [5]

In 2016, HackerRank surveyed programming language preferences across six industries: healthcare, social media, gaming and media, security, finance, and fintech. Analyzing the data gathered

from over 3000 coding challenges, they found out that while Java rules in three categories, there is one industry that favors Python significantly (it's twice as popular as the runner up, Java). [6]



Picture 1: 450+ coding challenges across 20+ companies in fintech (ref: HackerRank) [3]

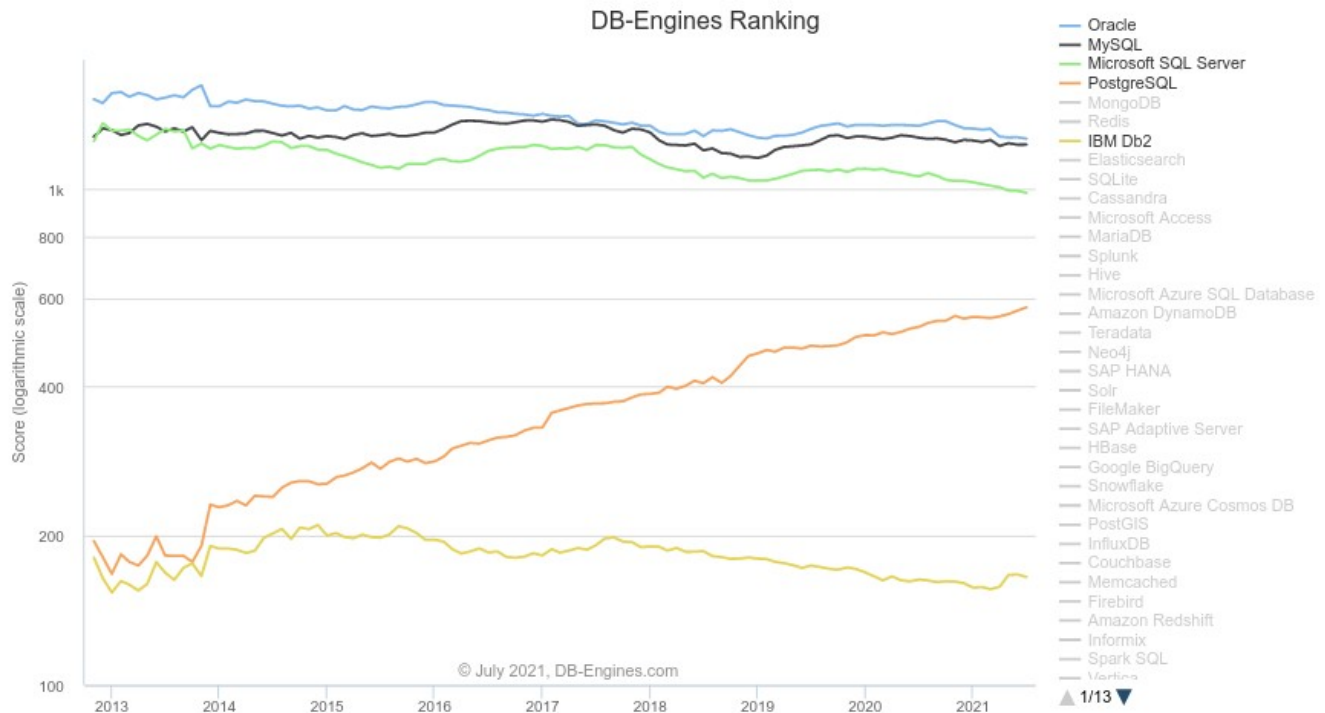
Therefore, based on numerous international experiences, it can be concluded that Python is a reliable tool for the development of fintech systems. Accordingly, the Python programming language was chosen to develop the Horizon project.

5.2 Database

When choosing the right database for banking systems, many bank executives immediately choose the Oracle database to develop their systems without research. Regardless of the consequences of choosing an Oracle database in terms of technical issues, development costs, system complexity, and so on. Research shows that this trend is changing around the world.

According to Gartner's The State of Open Source RDBMS, 2015, the OSDBMS market is worth \$562 million and has grown 31% year over year since 2013. [7] PostgreSQL was recognized as Database Management System (DBMS) of 2020, and is one of the most commonly used and loved database management systems by developers worldwide. [8] According to the Stack Overflow Developer Survey, PostgreSQL was the second most popular database in 2020. [9] There are many

other companies that use PostgreSQL, including Groupon (provides discounts on various products and services), Revolt (financial services), Trivago (one of the world's largest hotel search engines), and Accenture (global professional services leader). [10]



Picture 2: RDBMSs' Market Grow Up Until July 2021 [11]

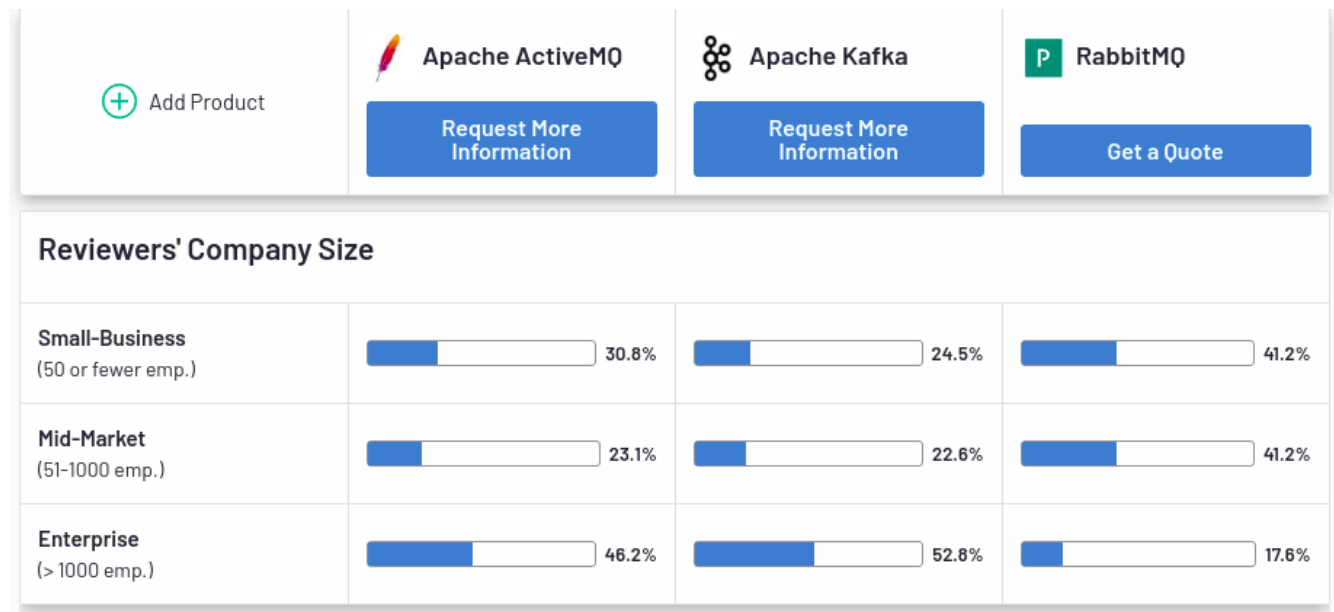
Therefore, due to the existence of appropriate educational resources, continuous growth of the consumer market, increasing epidemic, the existence of a worldwide network of programmers to review questions, open source and no need to obtain a license for new entries in the system, PostgreSQL has been chosen as database of Horizon project.

5.3 MQ System

It's critical to be certain that payments are sent once and once only. Using a message broker to handle these transactions' data offers assurance that payment information will neither be lost nor accidentally duplicated, provides proof of receipt, and allows systems to communicate reliably even when intermediary networks are down. [12]

The four popular and open source message queue systems are Apache Kafka, Apache MQ, RabbitMQ and Kestrel. Based on an some researches, ActiveMQ is another big player in the field with an impressive feature set. ActiveMQ is more in the RabbitMQ league than Kafka and like Kafka, it is written in Java. [13] According to G2.com comparison page, ActiveMQ is an easy to use MQ system which used for enterprise systems more than RabbitMQ (Not more than Kafka). [14] Apache Kafka is used to manage more complex messages such as streaming. But there is no need for this amount of

complexity and entanglement in the tools for payment systems and overwork in these types of systems should be avoided.



Picture 3: Company Size Comparison Between ActiveMQ, Kafka and RabbitMQ. [14]

Therefore, according to Horizon project policy regarding avoiding unnecessary complications and using free software, Apache ActiveMQ has been selected as the message queuing system. This system has good support for Python programming language and provides suitable communication interfaces for it.

5.4 In-memory Database and Cache

Redis is an open source (BSD licensed), in-memory data structure store, used as a database, cache, and message broker. Redis provides data structures such as strings, hashes, lists, sets, sorted sets with range queries, bitmaps, hyperloglogs, geospatial indexes, and streams. [15] According to monthly DB-Engines rankings, Redis is often the most popular key-value database. [16] It was voted most loved database in the Stack Overflow Developer Survey in 2017, 2018, 2019, and 2020 . [17] Therefore, the selected in-memory database for the Horizon project will be Redis.

6 Conclusion

In a software project, development tools are selected based on the needs, requirements and priorities of that project. Horizon project priorities are not based on similar projects in the field of fintech. Also, since this is a free software project, based on the views in the philosophy of free software, its development tools should also be free. The main promoter of this view is Richard Stallman, the founder of the concept of free software. Based on what was discussed in the previous sections, the technologies selected for the Horizon project are as follows:

Case	Selected Technologies
Programming Language and Development Kit	Python 3.x
Database	PostgreSQL 12
MQ System	Apache ActiveMQ
In-memory Database	Redis
Operating System for Development	Debian based Operation Systems (such as Ubuntu 20.04 LTS or later)

7 References

- [1] Dan Shafer. "[Python Success Stories](#)". python.org.
 - [2] Anmol Singh Sethi. "[Comparison of 10 Programming Languages](#)". Medium.com. May 15, 2020.
 - [3] Natalia Chrzanowska. "[9 Fintech Companies that Use Python in Software Development](#)". Netguru. Oct 3, 2018.
 - [4] Path Baroth. "[How To Use Python for FinTech Applications](#)". BoTree Technologies. January 04, 2021.
 - [5] Madhvi Mavadiya. "[Is Python storming ahead of Java in fintech?](#)". Finextra. March 18, 2020.
 - [6] Luiza, team member. "[Why are successful FinTech apps built with Python?](#)". MerixStudio. February 14, 2020.
 - [7] Ben Kerschberg. "[How Postgres and Open Source Are Disrupting The Market for Database Management Systems](#)". Forbes. Mar 8, 2016.
 - [8] Paul Andlinger, Matthias Gelbmann. "[PostgreSQL is the DBMS of the Year 2020](#)". DB-Engines. January 4, 2021.
 - [9] "[Databases Section - 2020 Developer Survey](#)". stackoverflow.com. February 2020.
 - [10] Jakub Romanowski. "[Which Major Companies Use PostgreSQL? What Do They Use It for?](#)". learnsql.com. May 19, 2020.
 - [11] "[DB-Engines Ranking - Trend Popularity](#)". DB-Engines. Until Jul 2021.
 - [12] "[Message Broker](#)". IBM Cloud Education. January 23, 2020.
 - [13] Yves Trudeau. "[Exploring Message Brokers: RabbitMQ, Kafka, ActiveMQ, and Kestrel](#)". DZone. June 3, 2014.
 - [14] "[Compare Apache ActiveMQ, Apache Kafka, and RabbitMQ](#)". G2.com. July 21, 2021.
 - [15] "[Index Page – About Redis](#)". Redis.io website. July 2021.
- Horizon Payment Switch

[16] “[DB-Engines Ranking of Key-value Stores](#)”. DB-Engines. July 2021.

[17] “[Most Loved, Dreaded, and Wanted Databases](#)”. Developer Survey Results 2019. stackoverflow.com.