Architecture Document

Balan Mihai-Stefan – Wealth Weaver Project

Contents

[1. Choices – Software 1](#_Toc191323573)

[2. Choices – Cloud&Network Architecture 2](#_Toc191323574)

# Choices – Technologies

In terms of software architecture, the following choices have been made, with their specific reasoning:

* Backend technology: C# .NET 8.0 – Object oriented, high-level with the ability to also be low-level when needed, large, trusted documentation base by Microsoft, maintained constantly, centralized core feature package, as opposed to the main contender, Java Springboot, versatile and perfectly suitable for the application described in the project plan.
  + This specific version of .NET is significantly faster than its predecessor, at an 18% speed increase and features Native Ahead-Of-Time (AOT) compilation, reducing deployment size significantly.
  + C# .NET can be used in Visual Studio, which is a very powerful IDE packed with developer tools to increase productivity.
* Design principles: The SOLID principles and the four core Object-Oriented Programming (OOP) principles—encapsulation, abstraction, inheritance, and polymorphism—are rigorously applied. To enhance code readability, maintainability, and expandability, composition is favored over inheritance, resulting in more modular and flexible code structures.
* Core architecture: The project will adopt a Service-Oriented Architecture (SOA) to ensure optimal performance and cost-efficiency. Main advantages:
  + Decoupling of services: SOA promotes the development of independent, loosely coupled services. This design principle allows for greater flexibility and easier maintenance of individual components.
  + Individual deployment: Each service can be developed, tested, and deployed separately, enhancing agility and reducing time-to-market for new features or updates.
  + Scalability: SOA enables independent scaling of components based on demand. This granular scalability allows for efficient resource allocation and cost optimization.
  + Reusability: Services can be reused across different parts of the application, reducing development time and promoting consistency.
  + Interoperability: SOA facilitates better integration between different systems and services, which is crucial for future expansions or third-party integrations.
  + Fault isolation: The decoupled nature of SOA improves system reliability by containing failures within individual services.
  + Cost effectiveness: following SOA principles, individual scaling of services based on demand will greatly reduce the cost compared to monolithic deployment.
* Cross-Service Messaging Broker: RabbitMQ – open-source decoupled messaging service. Features:
  + Asynchronous Communication: Enables services to communicate without waiting for immediate responses, improving system responsiveness.
  + Reliability: Ensures message delivery through features like acknowledgments and persistent storage.
  + Scalability: Supports high message throughput and allows horizontal scaling by adding nodes to the broker cluster.
  + Flexibility: Compatible with multiple protocols (AMQP, MQTT, STOMP) and programming languages.
  + Message Prioritization: Allows prioritization of messages, ensuring critical information is processed first.
  + Routing Capabilities: Offers powerful routing mechanisms like topic-based and header-based filtering for efficient message distribution.
  + High Availability: Supports clustering and replication for fault tolerance.
  + Monitoring Tools: Provides management plugins for performance tracking and troubleshooting

This was the preferred messaging broker over Apache Kafka, despite Kafka being objectively better at all the above-mentioned topics, due to the following factors:

* + Steeper learning curve: Learning Kafka components and configuration may endanger the timeplan and development process of the other core features
  + Unfit for current scenario: In the Wealth Weaver system, the low number of services results in an environment where the benefits in performance given by Kafka would not be visible.

# Choices – Cloud & Network Architecture

* Cloud provider chosen is Microsoft Azure: highly documented platform, many high-quality out-of-the-box resources to enhance application functions, native support for Kubernetes (AKS) and for FaaS architecture (Azure Function Apps). Extensive guides on cost-management and high-quality security measures through resources such as Bastion DMZ, Front Door, MSAL, etc.
* Core functionality: Function as a service was chosen. This decision is driven by the need for a highly scalable, cost-efficient, and event-driven architecture, perfectly suited to the application's core requirements. While SaaS solutions offer benefits for specific, non-core functionalities, FaaS provides a more compelling foundation for the dynamic processing and real-time responsiveness essential to Wealth Weaver's value proposition.
  + **Alignment with Event-Driven Architecture:** Core features like real-time spending insights, automated alerts, and immediate transaction categorization necessitate an event-driven model. FaaS excels in this area, triggering functions in response to specific events, such as exceeding a budget or a new transaction occurring. This ensures immediate action and responsiveness, which would be difficult to achieve with the more monolithic nature of typical SaaS offerings.
  + Granular and Cost-Effective Scalability**:** Wealth Weaver anticipates fluctuating workloads, with potential spikes during financial data synchronization or end-of-month budget updates. FaaS provides granular scalability, automatically scaling individual functions based on demand without requiring over-provisioning of entire servers. This "pay-per-use" model ensures cost efficiency by eliminating idle resource costs. SaaS solutions, on the other hand, often come with fixed pricing tiers that may not align perfectly with actual usage patterns, leading to potential overspending.
  + **Microservices Compatibility:** The application's microservices architecture benefits significantly from FaaS. FaaS allows core services, such as account aggregation, notifications, and spending analysis, to be decoupled into independent functions, simplifying updates and maintenance. This modularity provides greater flexibility compared to SaaS solutions, which typically offer less customization and control over underlying service components.
  + **Enhanced Control Over Security and Compliance:** While FaaS providers handle infrastructure security, encryption, and compliance, choosing FaaS for core logic allows greater control over the implementation of security measures and adherence to financial standards (e.g., PCI DSS). This can be a crucial advantage when dealing with sensitive financial data, offering more transparency and auditability compared to relying solely on SaaS providers.
  + **SaaS as a Complementary Approach**: While FaaS is the preferred choice for core functionalities, SaaS solutions are strategically integrated for non-core features where rapid deployment and readily available integrations are paramount. Examples include leveraging SaaS APIs for account aggregation (e.g., Plaid, Yodlee) and utilizing SaaS-based analytics platforms for budgeting and trend analysis. This hybrid approach allows Wealth Weaver to leverage the strengths of both FaaS and SaaS, optimizing development speed and overall system performance.