Part III: People

- Chapter 14. User Interfaces
  - Ownership Models in UI Development
    - Traditional Layered Architecture

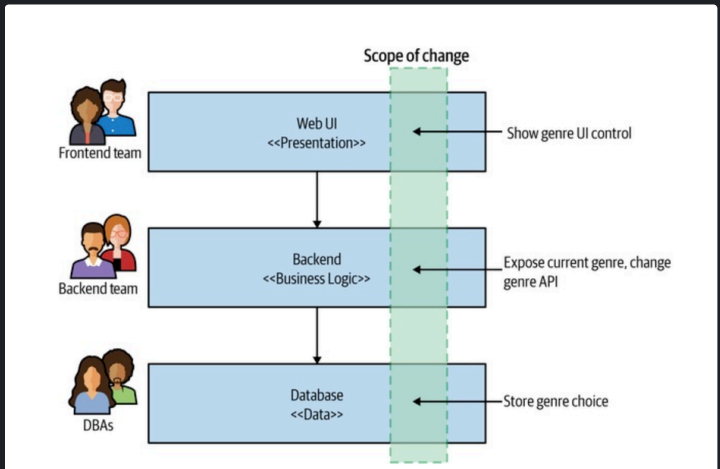

Figure 14-1. Making a change across all three tiers is more involved

      - **Structure:** Responsibility for the user interface (UI) layer is owned by a frontend team, while backend services are managed by another team
      - **Example:** Adding a simple control requires work from three different teams
      - **Impact:** This tiered organizational structure can slow down delivery due to the need for constant coordination and handoffs between teams
    - Preferred Ownership Model

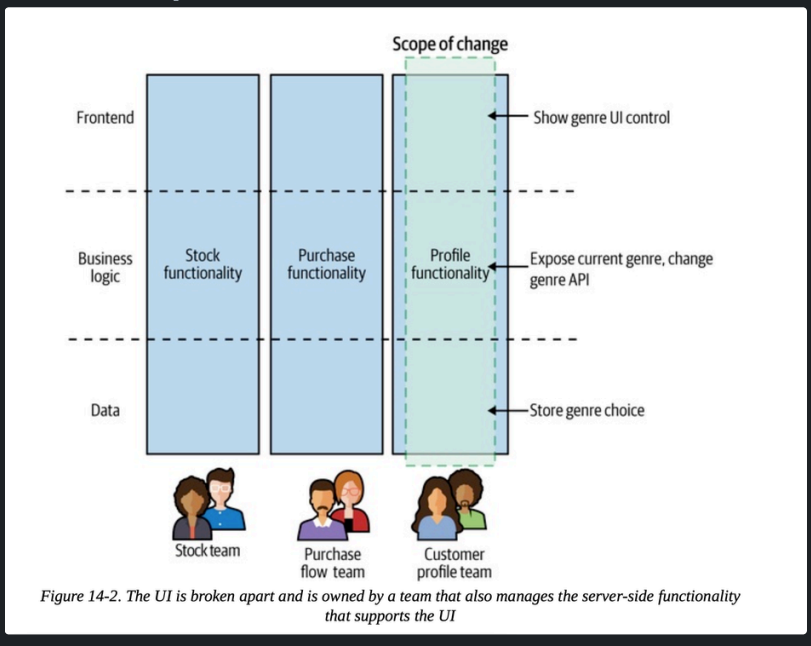

Figure 14-2. The UI is broken apart and is owned by a team that also manages the server-side functionality that supports the UI

      - **Structure:** The UI is broken apart and managed by a team that also manages the server-side components
      - A single team is responsible for all changes, enabling faster modifications
      - Full ownership encourages direct contact with end users, maintaining focus on user needs
    - Current Common Practice: Despite the drawbacks, dedicated frontend teams are still common in companies using microservices

        - Reasons for Dedicated Frontend Teams
            - Scarcity of Specialists:
                - Specialized skills in interaction and graphical design, as well as technical know-how, are hard to find
                - Specialists are often grouped together to focus on their expertise
            - Consistency
                - A single frontend team can ensure a consistent look and feel across the UI
                - Consistent controls and design elements create a cohesive user experience
            - Technical Challenges
                - Some UI technologies, particularly single-page applications (SPAs), are challenging to break apart
                - SPAs serve the entire application within a single web page, making decomposition difficult
                - Frameworks like Angular, React, and Vue are used for sophisticated UIs
    - **Pattern: Monolithic Frontend**
        - Overview:
            - The monolithic frontend pattern involves a single, unified UI that handles all state and behavior, while making calls to backing microservices to fetch data and perform operations
            - **Example:** A screen displaying album details pulls data from the Album microservice and promotional data from the Promotions microservice, updating the UI with the received JSON data.
    - **Pattern: Micro Frontends**
    - **Pattern: Page-Based Decomposition**
    - **Pattern: Widget-Based Decomposition**
    - **Pattern: Central Aggregating Gateway**
    - **Pattern: Backend for Frontend (BFF)**
- Chapter 15. Organizational Structures
    - **Loosely Coupled Organizations**
    - **Conway’s Law**
        - Overview: Conway’s Law states that the design of a system reflects the communication structure of the organization that creates it. This concept highlights the link between organizational structure and system architecture.
        - Definition: Melvin Conway's 1968 paper stated, "Any organization that designs a system will inevitably produce a design whose structure is a copy of the organization’s communication structure
        - Reinforcement: A loosely coupled organization results in a loosely coupled architecture. Conversely, a tightly coupled organization produces a tightly coupled architecture. This alignment is crucial for achieving the benefits of a microservice architecture.
        - Evidence
            - **Loosely and Tightly Coupled Organizations:** Research showed that loosely coupled organizations (like open source communities) created more modular systems than tightly coupled organizations (like commercial firms).
            - **Windows Vista:** Microsoft’s study on Windows Vista revealed that organizational metrics (e.g., number of engineers on a component) were significant predictors of error-proneness, more so than traditional software quality metrics
            - **Amazon and Netflix:** Both companies exemplify the alignment of organizational structure with system architecture
                - **Amazon:** Adopted the "two-pizza team" concept, aiming for small teams (8-10 people) owning and operating their systems. This led to the creation of Amazon Web Services to support team independence
                    - The "two-pizza team" concept is an organizational principle popularized by Amazon's CEO, Jeff Bezos. The idea is that no team should be so large that it cannot be fed with two pizzas. This concept is used to optimize team size for better productivity, communication, and agility
                    - Optimal Team Size: The idea suggests that the ideal team size is around 6 to 10 people, assuming that two large pizzas can feed this many people. This size is considered small enough to maintain effective communication and coordination but large enough to have diverse skills and perspectives

        - Autonomy and Ownership: Each team is responsible for a specific piece of the product or service. They have end-to-end ownership, meaning they handle everything from development to deployment and maintenance. This reduces dependencies on other teams and allows for faster decision-making and problem-solving
        - Reduced Communication Overhead: Smaller teams have fewer communication channels, which simplifies coordination and reduces the chances of miscommunication. In a large team, the number of communication pathways increases exponentially, making coordination more complex
        - Agility and Speed: Smaller, autonomous teams can move faster and adapt more quickly to changes. They can iterate on their product or service more frequently, leading to faster innovation and quicker responses to market demands or customer feedback
        - Empowerment and Accountability: By giving teams full ownership of their work, members are more empowered and accountable for their outcomes. This can lead to higher motivation and a stronger sense of responsibility.
        - The two-pizza team concept is a practical approach to team organization that emphasizes small, autonomous, and empowered teams. By optimizing team size and promoting ownership and accountability, organizations can enhance productivity, agility, and innovation. This approach has been instrumental in Amazon's success and is widely adopted in various industries to foster effective and dynamic team structures
      - **Netflix:** Structured around small, independent teams to ensure services were independent and optimized for rapid changes. Teams whose services interacted closely were seated together to facilitate communication
    - Conclusion: Conway’s Law emphasizes the importance of aligning organizational structures with desired system architectures. Effective communication and small, autonomous teams are key to achieving modular and efficient systems, as demonstrated by leading companies like Amazon and Netflix
- Chapter 16. The Evolutionary Architect