Balancing control and autonomy in software platform ecosystems

INF5750

2017

Figures and citations are taken from: Ghazawneh, Ahmad, and Ola Henfridsson. "Balancing platform control and external contribution in third-party development: the boundary resources model." *Information Systems Journal* 23, no. 2 (2013): 173-192.

Contents and learning outcome of the lecture

- What are boundary resources
- Some important concepts
- The dynamics between platform owners and third party developers through boundary resources drivers
- Some implications

Main components of a software platform

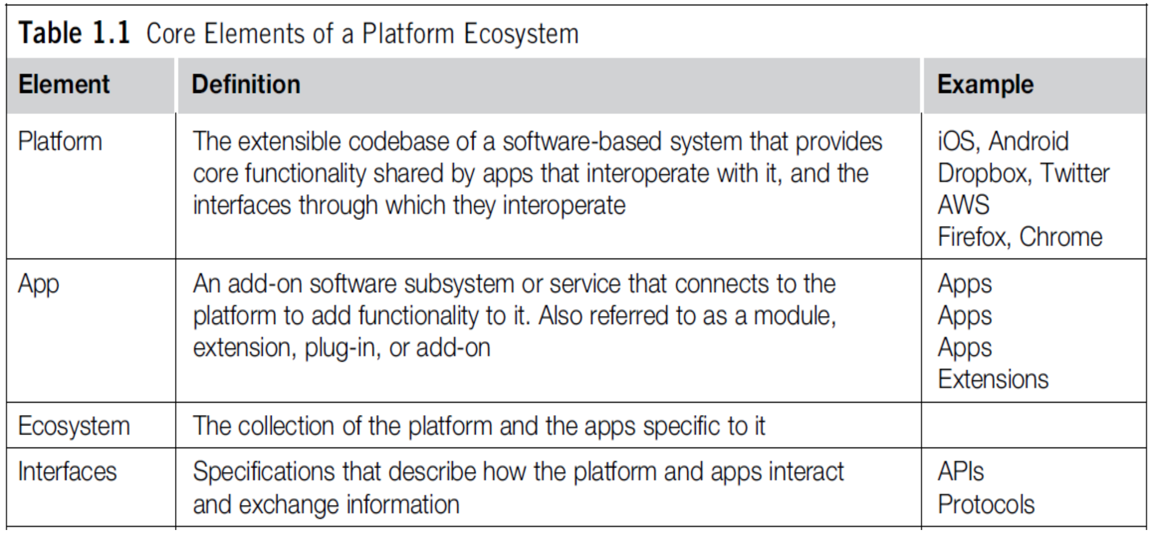


Table 1.1 Core Elements of a Platform Ecosystem

Element	Definition	Example
Platform	The extensible codebase of a software-based system that provides core functionality shared by apps that interoperate with it, and the interfaces through which they interoperate	iOS, Android Dropbox, Twitter AWS Firefox, Chrome
App	An add-on software subsystem or service that connects to the platform to add functionality to it. Also referred to as a module, extension, plug-in, or add-on	Apps Apps Apps Extensions
Ecosystem	The collection of the platform and the apps specific to it	
Interfaces	Specifications that describe how the platform and apps interact and exchange information	APIs Protocols

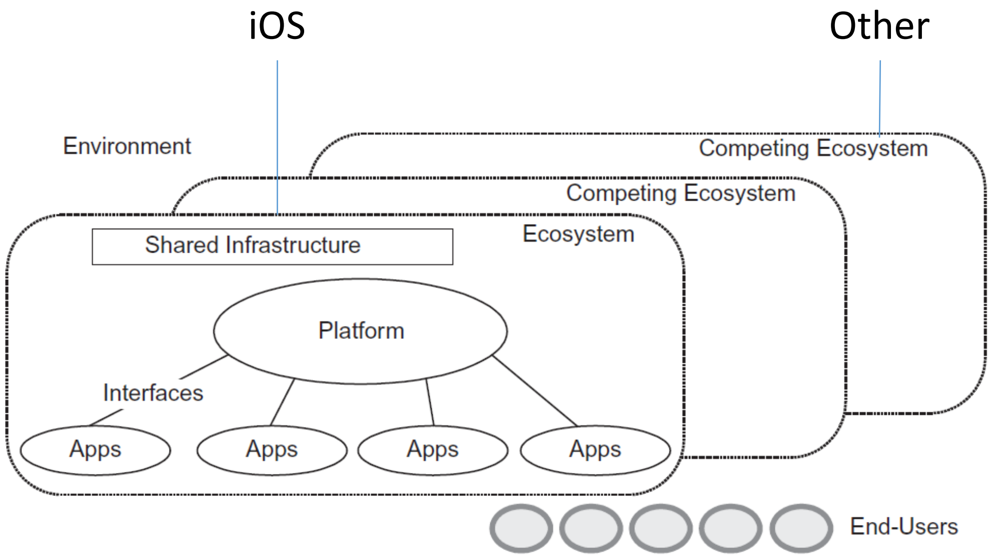
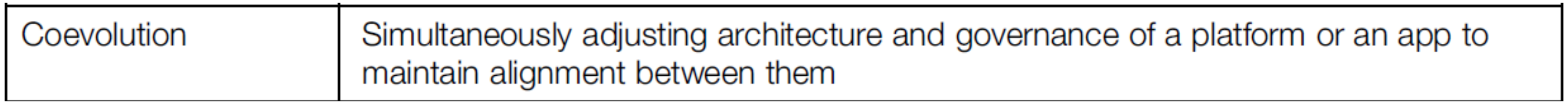


FIGURE 1.1

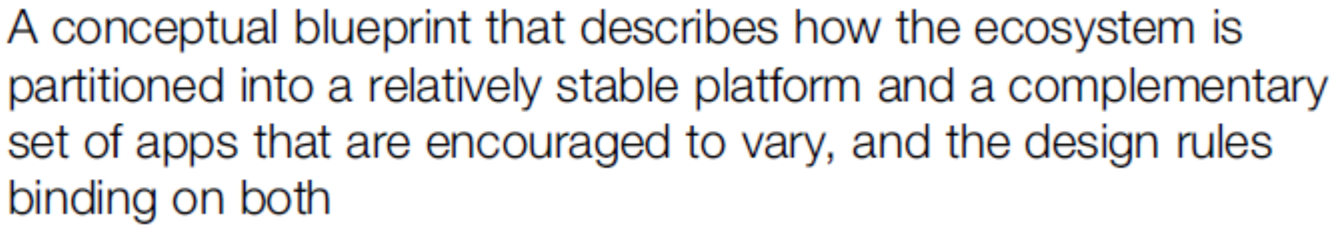
Elements of a platform ecosystem.



Coevolution	Simultaneously adjusting architecture and governance of a platform or an app to maintain alignment between them

Evolution of platform ecosystems

- Architecture: Structure



A conceptual blueprint that describes how the ecosystem is partitioned into a relatively stable platform and a complementary set of apps that are encouraged to vary, and the design rules binding on both

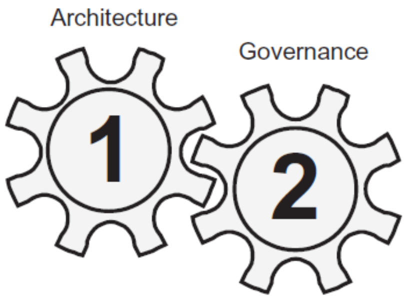
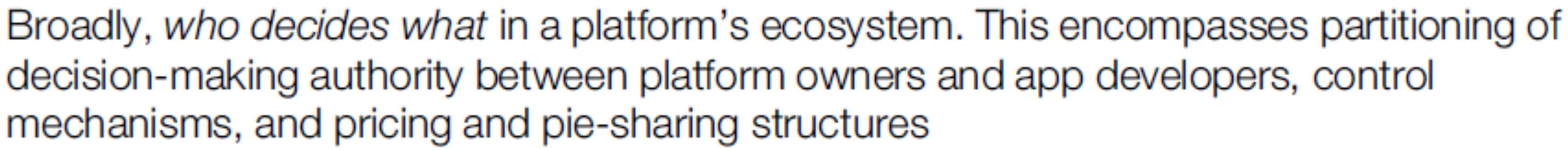


FIGURE 2.19

Architecture and governance are the two gears of evolution of a platform ecosystem.

- Governance: Process and rules



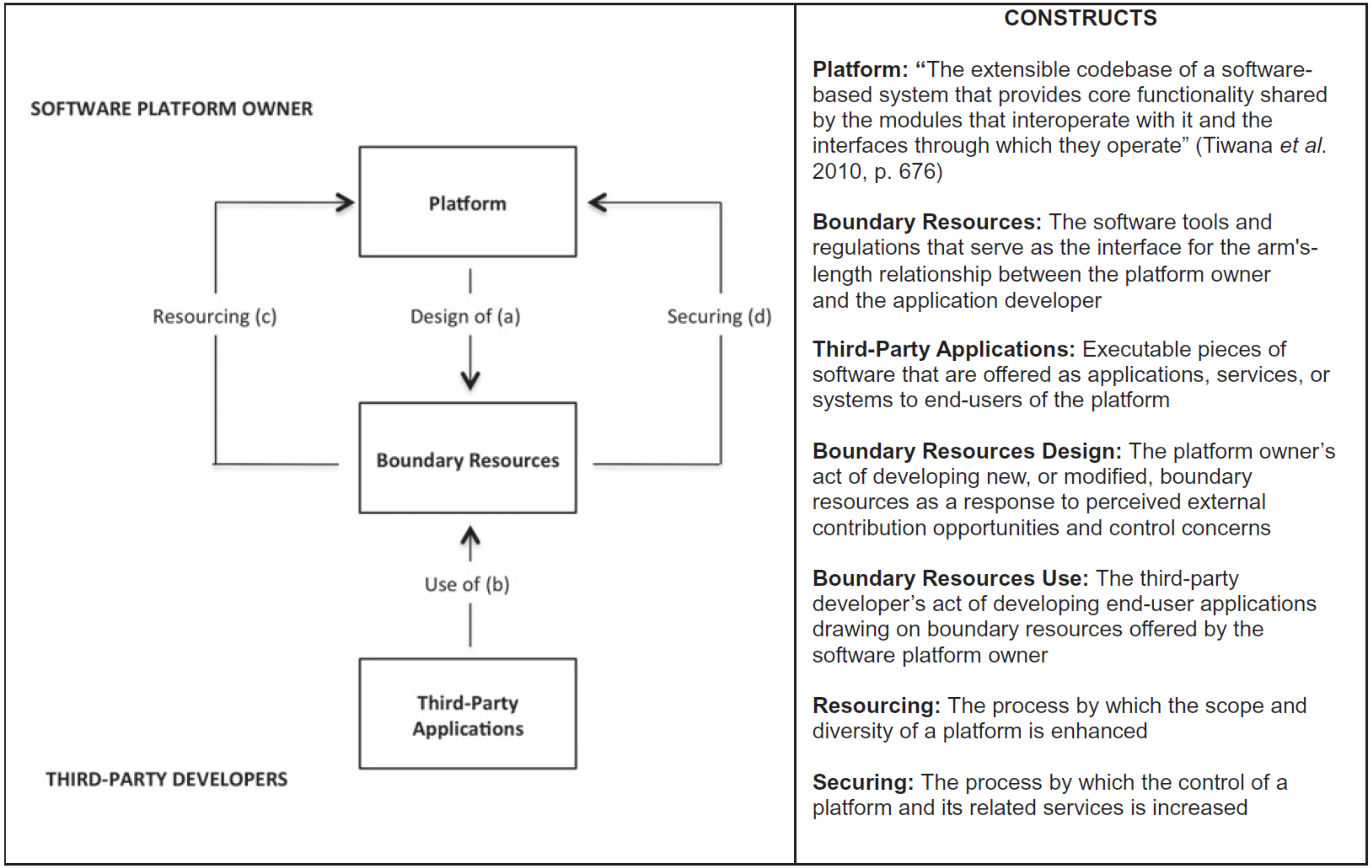
Broadly, *who decides what* in a platform's ecosystem. This encompasses partitioning of decision-making authority between platform owners and app developers, control mechanisms, and pricing and pie-sharing structures

- Evolution: «… *the interplay between its irreversible architecture and how it is governed*."

What are boundary resources?

- «… *the software tools and regulations that serve as the interface for the arm's-length relationship between the platform owner and the application developer.*"
  - Enables the relation between platform and third party
- APIs, Software development kits, regulations
- A question of balancing control and autonomy between platform owner and third party contributors
- What is the role of boundary resources in this process?
  - How to understand the dynamics between platform owner and third party contributors

Concepts



CONSTRUCTS

Platform: "The extensible codebase of a software-based system that provides core functionality shared by the modules that interoperate with it and the interfaces through which they operate" (Tiwana *et al.* 2010, p. 676)

Boundary Resources: The software tools and regulations that serve as the interface for the arm's-length relationship between the platform owner and the application developer

Third-Party Applications: Executable pieces of software that are offered as applications, services, or systems to end-users of the platform

Boundary Resources Design: The platform owner's act of developing new, or modified, boundary resources as a response to perceived external contribution opportunities and control concerns

Boundary Resources Use: The third-party developer's act of developing end-user applications drawing on boundary resources offered by the software platform owner

Resourcing: The process by which the scope and diversity of a platform is enhanced

Securing: The process by which the control of a platform and its related services is increased

Ecosystem dynamics

- Self-resourcing
  - -> third-party developers’ act of developing new boundary resources as a response to perceived limitations in existing boundary resources
  - -> developers’ own initiatives to resource the platform in ways that benefit their application development
    - Examples: Jailbreaking of Apple iPhone – other types of apps and functionality made possible than what the iOS platform enabled -> shift in Apples strategy and interfaces
- Regulation-based securing
  - -> platform owners’ act of exercising control over the platform and its related services
  - -> control actions that rely on regulative measures rather than technical restrictions
    - Examples: Apple’s regulations of which apps that were allowed to be a part of the app store. Not an architectural solution, but by governance

Ecosystem dynamics

- Diversity resourcing
    - -> deliberate action taken by a platform owner to diversify the platform in a way that stimulates new application areas
    - -> how third-party development can enable a platform owner to transform its enterprise beyond its traditional industrial settings
        - Examples: Growing criticism about closed platform -> Apple iPhone 3G – new features – 1000 APIs
- Sovereignty securing
    - -> actions taken by a platform owner to maintain control of the platform’s evolution and avoid becoming a substitute platform for application developers
    - -> responses of platform owners when other platform owner’s attempts to scale their own platform and ecosystems
        - Examples: Apple restricting license agreements hindering other platforms to be injected – hindering for instance Adobe packager for iPhone translating apps developed in their flash developer tool to be translated to iOS apps.

Some possible impications

- From understanding interfaces as standards that ones have to comply with -> appreciating BR design as strategising
- Enables micro-level analysis to be able to understand software platform ecosystem evolution better (than previous research)
- Acknowledging the tensions and dynamics between resourcing and securing, and between platform owner and thrid party contributors