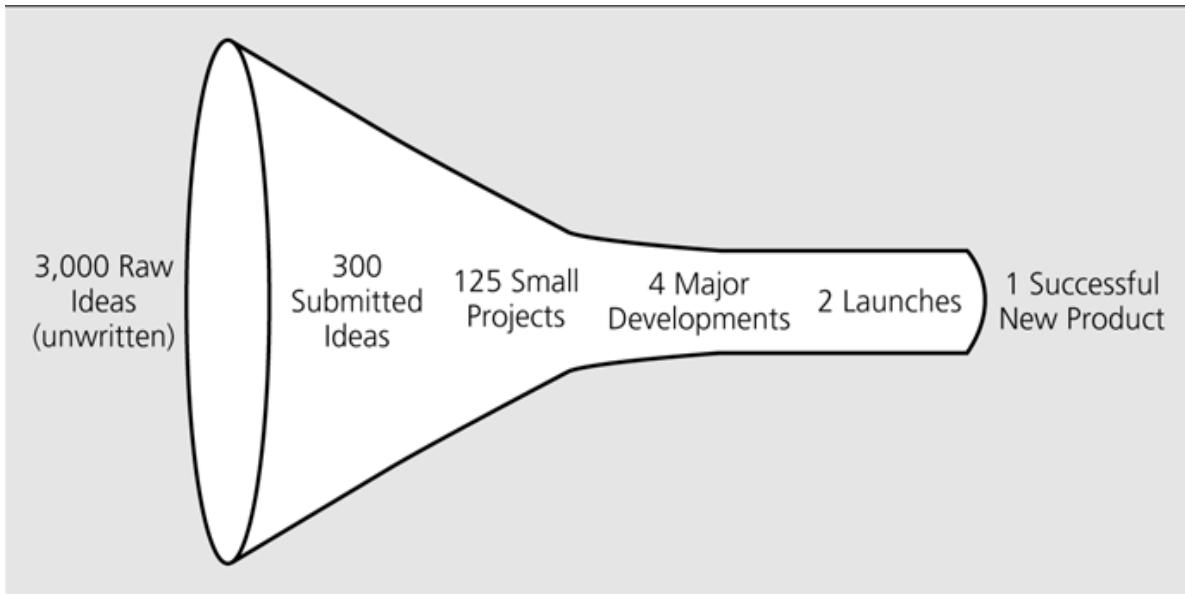


Evolution of innovation by companies: Traditional model

- This is a simplistic model assuming:
 - Simple one-way flow – left to right (it's not usually this simple)
 - All activities inside a single company (no in-flows, no out-flows)



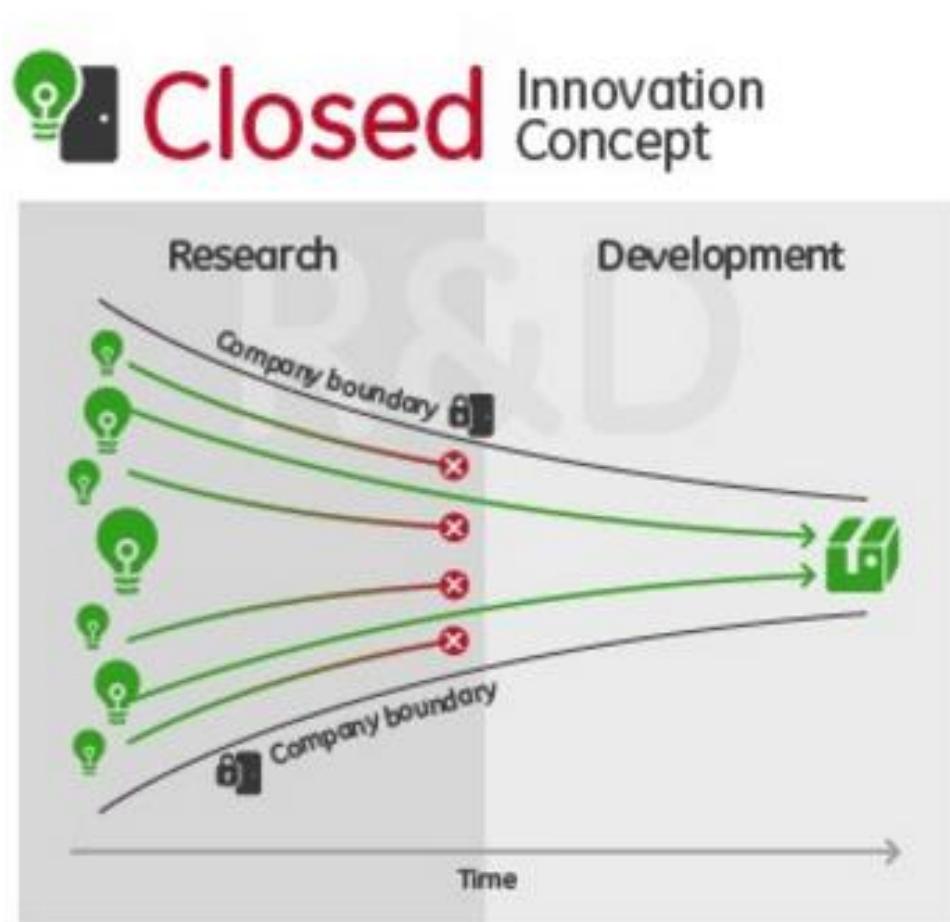
Source – Schilling, 2013

Evolution of innovation by companies:

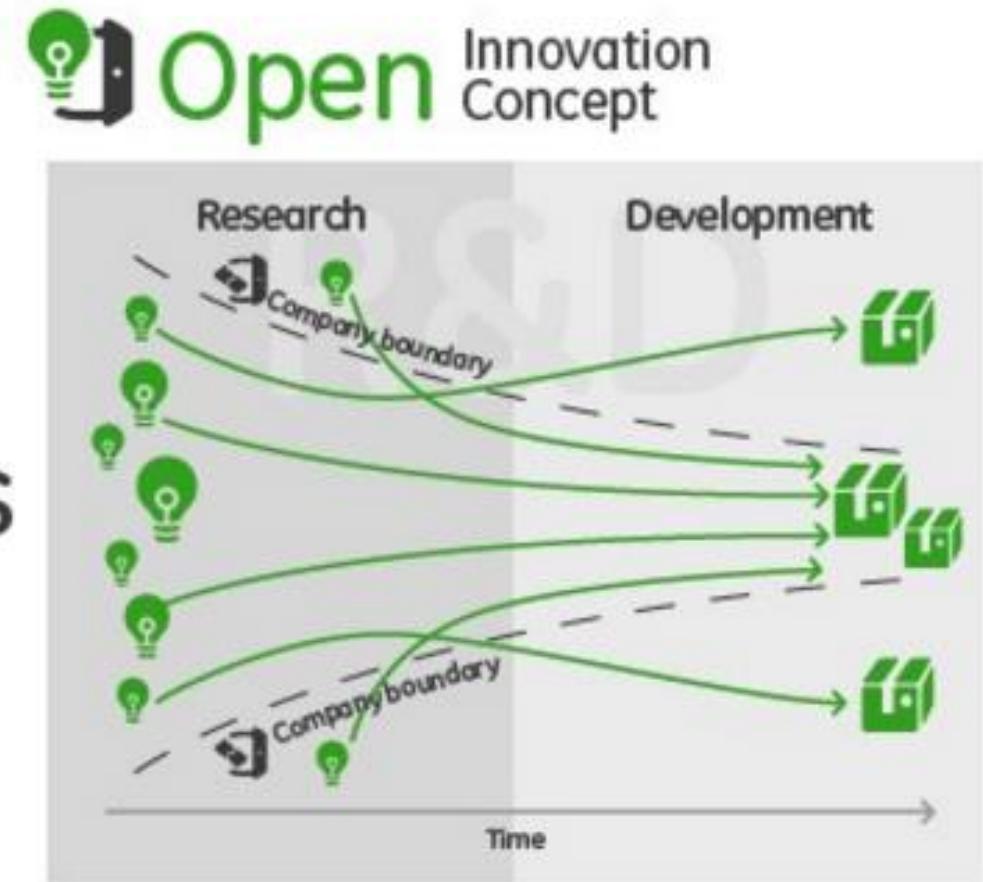
Some trends in the late 20th century

- More mobility of workers between companies
- More outsourcing of work
- Globalisation (more working across countries)
- Better information and communication technologies (e.g., email, web)
- Availability of venture capital funding allowing small companies to grow quickly (even without revenue)
- Easier to create and build new technology companies
- So more opportunities for collaborative innovation

The classic innovation funnel: “Closed innovation”



VS



Types of open innovation

1. Outside-in process:

- “Enriching the company’s own knowledge base through the integration of suppliers, customers, and external knowledge sourcing”. E.g., Microsoft acquired GitHub

2. Inside-out process:

- “Earning profits by bringing ideas to market, selling IP, and multiplying technology by transferring ideas to the outside environment.”, E.g., Qualcomm's Licensing Model

3. Coupled process:

- “co-creation with (mainly) complementary partners through alliances, cooperation, and joint ventures during which give and take are crucial for success.”. E.g., BMW, Intel, and Mobileye Collaboration

Source: Enkel, Gassmann and Chesbrough (2009)

Some benefits of open innovation

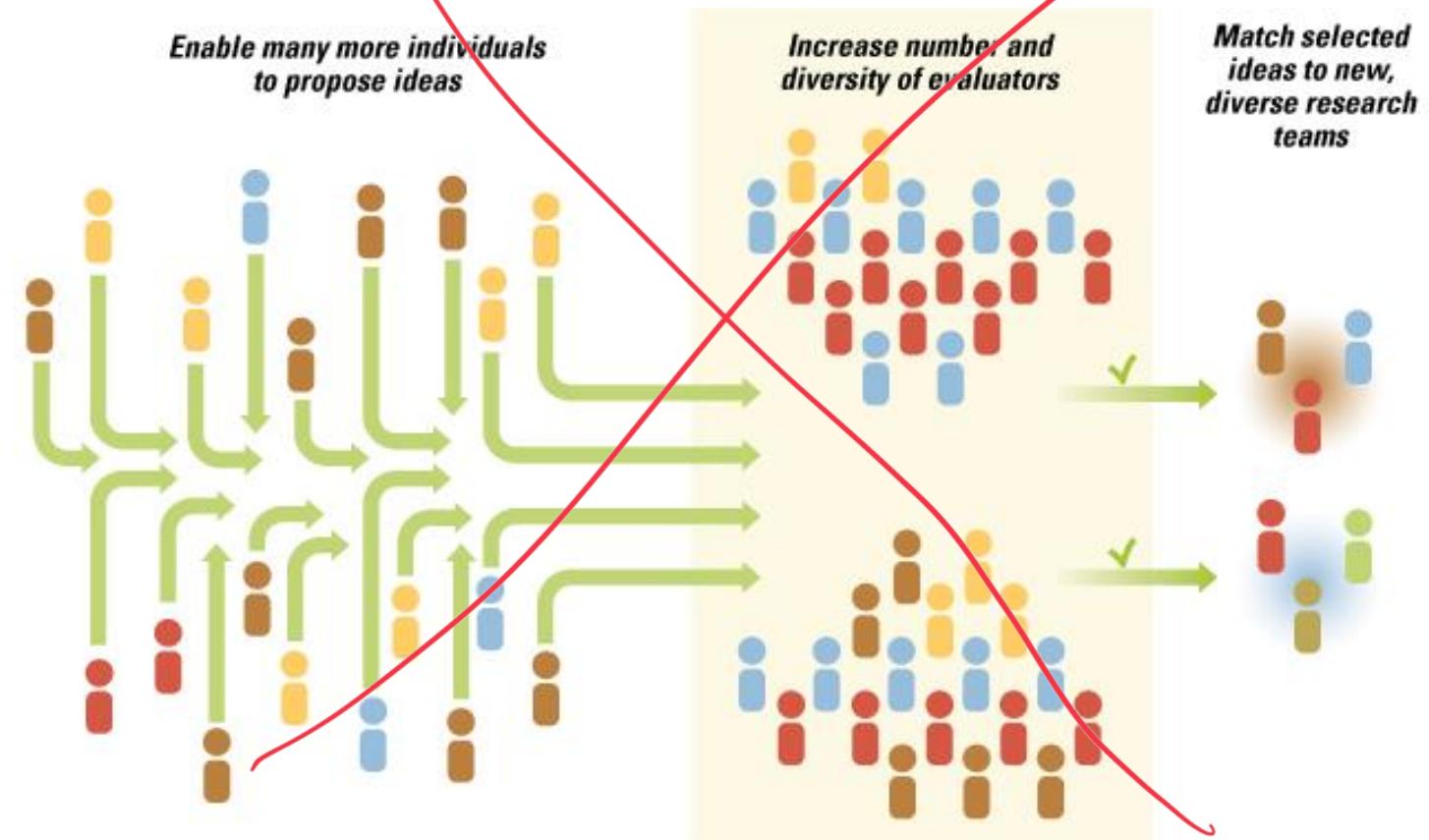
-  Larger base of ideas to draw from for innovation
 - “Not all of the smart people work for us” (Bill Joy from Sun Microsystems)
-  Existing third-party technology can be used, reducing risk and cost of development
-  Identification of new business opportunities with collaborators
-  Share risks and pool resources with other companies
-  Can be lower cost than large R&D departments

Risks of open innovation (compared to closed innovation)

-  Lack of control
 - Will usually not have as tight control of external resources as internal ones
-  Higher complexity of managing innovation
 - Need to manage external relationship, intellectual property, confidentiality etc
-  Higher coordination costs
 - May cost to coordinate external resources
-  Possible loss of own capability over time
 - If are not using and building a capability but relying on others
-  Possible loss of competitive advantage compared to others
 - If allow others to build skills in area important to your business, they can sell their expertise to your competitors (contracts can help address the risk)

Distributed innovation

- “a system in which innovation emanates not only from the manufacturer of a product but from many sources including users and rivals”
- Eric von Hippel (1988) paraphrased by Carliss Baldwin (2012)



<http://sloanreview.mit.edu/article/experiments-in-open-innovation-at-harvard-medical-school/> (Mar'25)

Enabling distributed innovation: **Modularity**

- In software engineering, modularity refers to how much a software/Web application may be divided into smaller modules. Software modularity indicates that the number of application modules can serve a specified business domain.

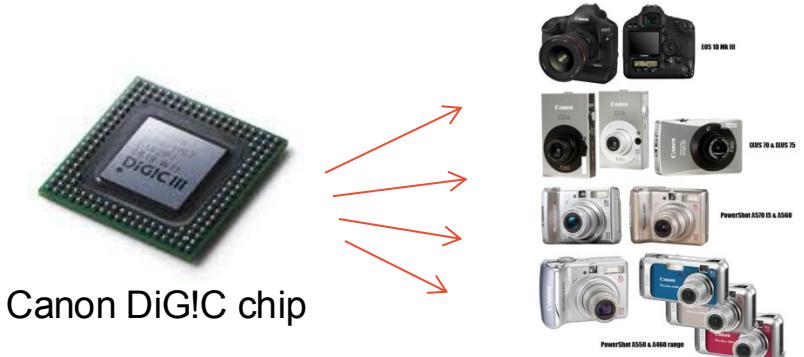
<https://www.techopedia.com/definition/24772/modularity>

Enabling distributed innovation: **Modularity**

- Products may be modular at:
 - **User level** e.g. Firefox add-ons, Microsoft Office plug-ins, Smartphone Apps
 - **Producer (company) level** e.g. Software products based on a company's platforms
 - **Industry level** e.g. Each component of a PC made by different company

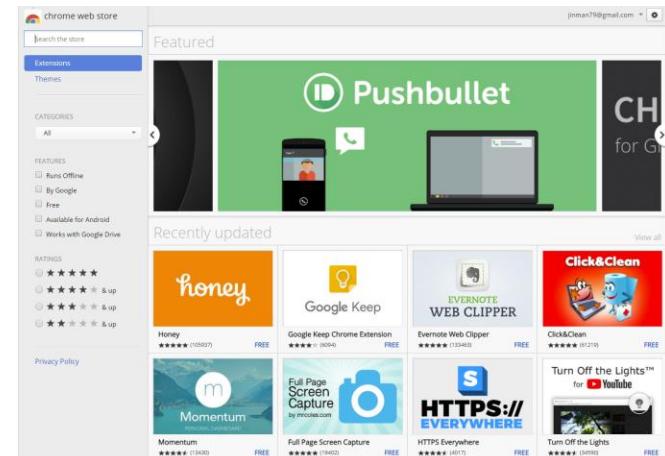
Product Platforms

- Concept became popular in the 90s – used for reusable components/design frameworks
- Foundation of components around which a company builds related products
- Also known as “product family engineering”
- Platforms make it possible for companies to:
 - Have a rich line-up of different products with the same core functions
 - At different price-points
 - For different customer types
 - To do so efficiently through re-use of a common platform



Canon DiG!C chip

<http://www.reghardware.com/>

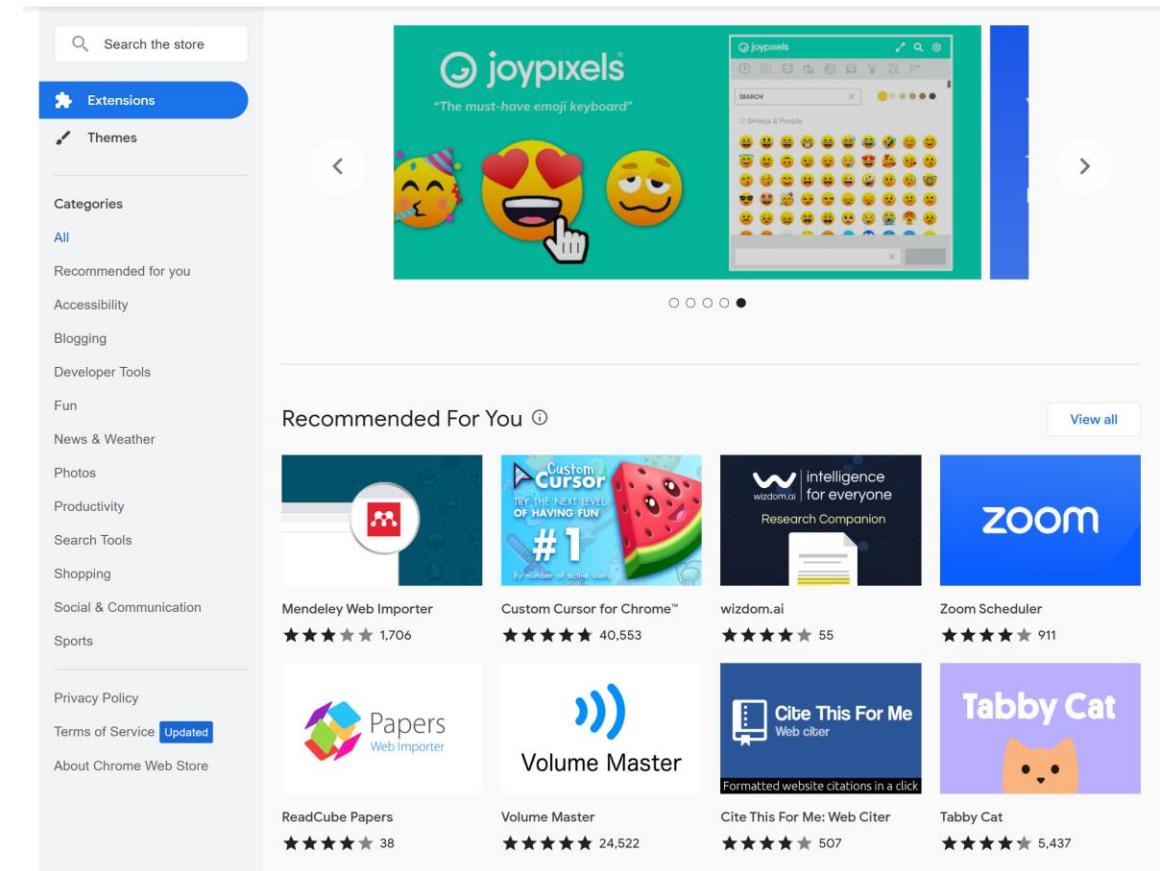


Some ways in which companies provide IT product platforms

- Make source code available:
 - Allows external innovators to modify the software for their own needs
 - E.g., Core Java platform
- Provide toolkit (software and documentation):
 - Allows external innovators to write software based on the toolkit
 - E.g., SAP XML Toolkit for Java
- Provide plug-in/add-on support in software:
 - Allows external innovators to customise software without access to source code
 - E.g., Google Extensions
- Provide a complete product platform for external innovation
 - Allows external innovators to write rich and varied applications on the platform
 - E.g., Android and iPhone app architectures
- Provide live data/functionality via an application programming interface (API)
 - Allows external innovators to build new services using the data
 - E.g., Facebook API

Example of a Product Platforms – Google Extensions

- Extensions are small software programs that **customise the browsing experience**. They enable users to tailor Chrome functionality and behavior to individual needs or preferences.
- They are built on web technologies such as HTML, JavaScript, and CSS.



[Chrome Web Store - Extensions \(google.com\) \(Mar'25\)](https://chrome.google.com/webstore/category/extensions)

Product Platforms: Benefits

- For external product platform:
 - Can be made available externally, leading to new businesses, and new business models
- For internal product platform:
 - Reuse technology component in multiple products leading to:
 -  Faster development time so gets to market sooner
 -  Lower effective cost (as spread over multiple products)
 -  Higher adaptability and ‘evolve-ability’
 -  Innovative aspects of the platform can benefit a range of products
 -  Application development on platform can focus on innovative value-add

Web APIs

- Interfaces for web-based services to interact (usually RESTful APIs)
- Enable modularity on the web
- Used e.g.:
 - Maps
 - Payment
 - Messaging
- Becoming the underlying infrastructure for a lot of automation

Image: developer.google.com

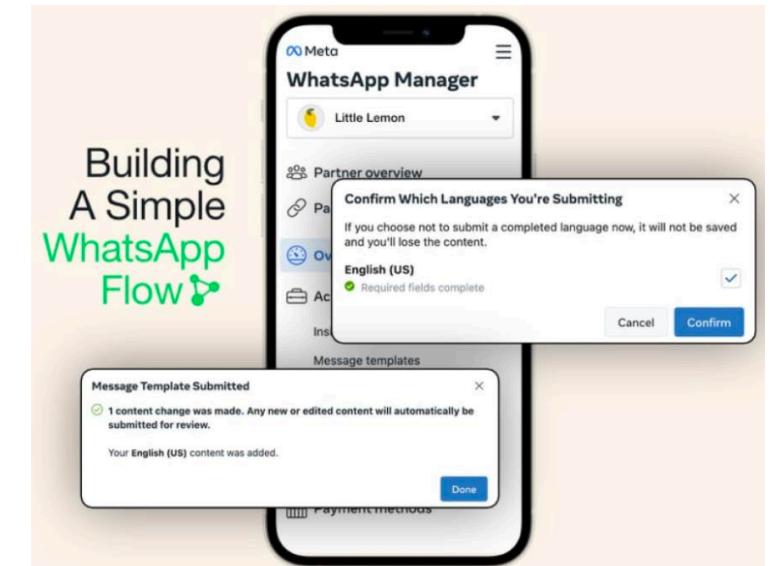
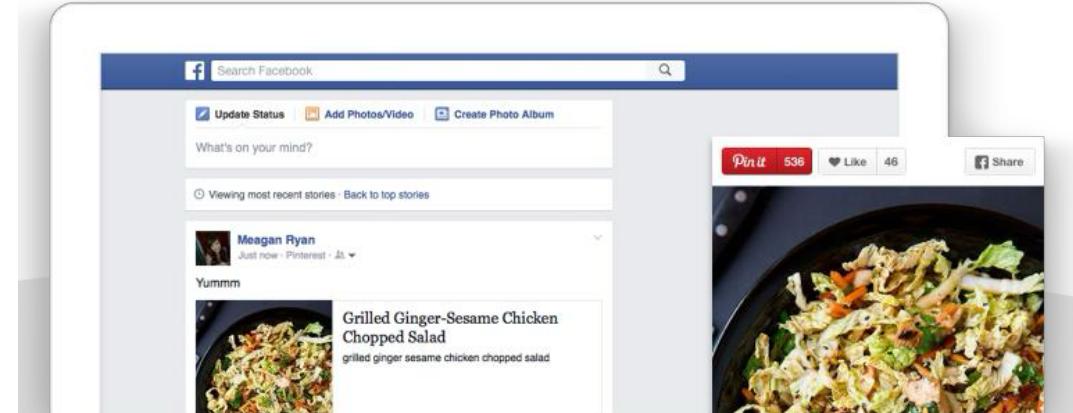


Image: facebook.com/developer

API Business models

- Models can be: Free /Developer Pay / Developer Gets Payed / Indirect
- **API as a product:** This category implies that the API has a specific money-making goal or serves as a significant or single source of income for the company. By definition, APIs in this category must provide value that is easy to monetize, and is highly competitive or unique
- **API enhancing existing product:** A majority of monetized APIs fall into this category. With the main money-making responsibility assigned to another part of the business, API providers have a greater set of business model options, ranging from direct pay-to-play to indirect, commission-based compensation
- **API promoting existing product:** Designed to solidify the market position, APIs in this category are often offered for free, and work to attract interest and traffic to the API provider.

<https://www.epam.com/insights> (Mar'25)

[A Quick Guide to Business Models for APIs](#) (Mar'25)