Analyzing Business Domains

In this chapter, we will learn domain-driven design tools for analyzing a company’s

business domain and its structure: its **core**, **supporting**, and **generic subdomains.**

# What is a Business Domain?

A business domain **defines a company’s main area of activity**. Generally speaking, it’s

the service the company provides to its clients. For example:

• FedEx provides courier delivery.

• Starbucks is best known for its coffee.

A company can operate in multiple business domains. For example, Amazon provides

both retail and cloud computing services. Uber is a rideshare company that also provides food delivery and bicycle-sharing services.

# What is A Subdomain?

To achieve its business **domain’s goals** and targets, a company has to **operate in multiple**

subdomains.

A subdomain is a fine-grained area of business activity.

All of a company’s subdomains form its business domain: the service it provides to its customers.

Implementing a single subdomain is not enough for a company to succeed; it’s just

one building block in the overarching system.

The subdomains have to interact with each other to achieve the company’s goals in its business domain. For example, Starbucks may be most recognized for its coffee, but building a successful coffeehouse chain requires more than just knowing how to make great coffee. You also have to buy or rent real estate at effective locations, hire personnel, and manage finances, among other activities. None of these subdomains on its own will make a profitable company. All of them together are necessary for a company to be able to compete in its business domain(s).

subdomains bear **different strategic/business values**. Domain-driven design distinguishes between three types of subdomains: core, generic, and supporting. Let’s see how they differ **from a company strategy point of view.**

## Core Subdomains

A core subdomainis what a company does differently from its competitors. This may

involve inventing new products or services or reducing costs by optimizing existing processes.

Let’s take Uber as an example. Initially, the company provided a novel form of transportation: ridesharing. As its competitors caught up, Uber found ways to optimize

and evolve its core business: for example, reducing costs by matching riders heading in the same direction.

To maintain a competitive advantage, core subdomains involve inventions, smart optimizations, business knowhow, or other intellectual property.

### Core Subdomains are Naturally Complex

A core subdomain that is simple to implement can only provide a short-lived competitive advantage. Therefore, core subdomains are naturally complex.

There should be high entry barriers for a company’s core business; it should be hard for

competitors to copy or imitate the company’s solution.

### Core Subdomains are the Source of Competitive Advantage and are not Necessarily Technical

Not all business problems are solved through algorithms or other technical solutions. A company’s competitive advantage can come from various sources.

Consider a jewelry maker selling its products online. The online shop is important, but it’s not a core subdomain. The jewelry design is. **The company can use an existing off-the-shelf online shop engine, but it cannot outsource the design of its jewelry**. The design is the reason customers buy the jewelry maker’s products and remember the brand.

### Core Domains vs Core Subdomains

Core subdomains are also called core domains. in the original domain-driven design book, Eric Evans uses “core subdomain” and “core domain” interchangeably.

Although the term “core domain” is used often, I prefer to use “core subdomain” for a number of reasons. First, it is a *subdomain,* and I prefer to avoid confusion with *business domains*. Second, as we will see that it’s not uncommon for subdomains to evolve over time and change their types. For example, a core subdomain can turn into a generic subdomain. Hence, saying that “a *generic* subdomain has evolved into a *core* subdomain” is more straightforward than saying “a generic *subdomain* has evolved into a core *domain.*”(*I think they made a typo here but the point is still there.*)

## Generic Subdomains

*Generic subdomains* are **business activities that all companies are performing in the same way**. **Like core subdomains, generic subdomains are generally complex and**

**hard to implement**. **However, generic subdomains do not provide any competitive**

**edge for the company**.

There is no need for innovation or optimization here: battletested implementations are widely available, and all companies use them. **For example**, most systems need to **authenticate and authorize their users**. Instead of inventing a proprietary authentication mechanism, it makes more sense to use an existing solution. Such a solution is likely to be more reliable and secure since it has already been tested by many other companies that have the same needs.

Going back to the example of a jewelry maker selling its products online, jewelry design is a core subdomain, but the online shop is a generic subdomain. Using the same online retail platform—the same generic solution—as its competitors would not impact the jewelry maker’s competitive advantage.

## Supporting Subdomains

As the name suggests, ***supporting subdomains* support the company’s business**. However, contrary to core subdomains, (*and I add: similar to generic subdomains*)supporting subdomains do not provide any competitive advantage.

For example, consider an online advertising company whose core subdomains include matching ads to visitors, optimizing the ads’ effectiveness, and minimizing the cost of ad space.

However, to achieve success in these areas, **the company needs to catalog its creative materials**. The way the company stores and indexes its physical creative materials, such as banners and landing pages, does not impact its profits. There is nothing to invent or optimize in that area. On the other hand, the creative catalog is essential for implementing the company’s advertising management and serving systems. That makes the content cataloging solution one of the company’s supporting subdomains.

The distinctive characteristic of supporting subdomains is the complexity of the solution’s

business logic. Supporting subdomains are simple. Their business logic resembles mostly data entry screens and ETL (extract, transform, load) operations; that is, the so-called CRUD (create, read, update, and delete) interfaces. These activity areas do not provide any competitive advantage for the company, and therefore do not require high entry barriers.

*I add: generic subdomains are infrastructures that are needed for the business to operate, they can be used across different businesses with little to no customizations and there are ready-to-use solutions for them. supporting subdomains on the other hand are specific to that business, yet they don’t provide the same value as the core subdomains. they are not complex. these are just the things that has to be done yet there is no ready-to-use solution for them.*

# Comparing Subdomains

**…page7**