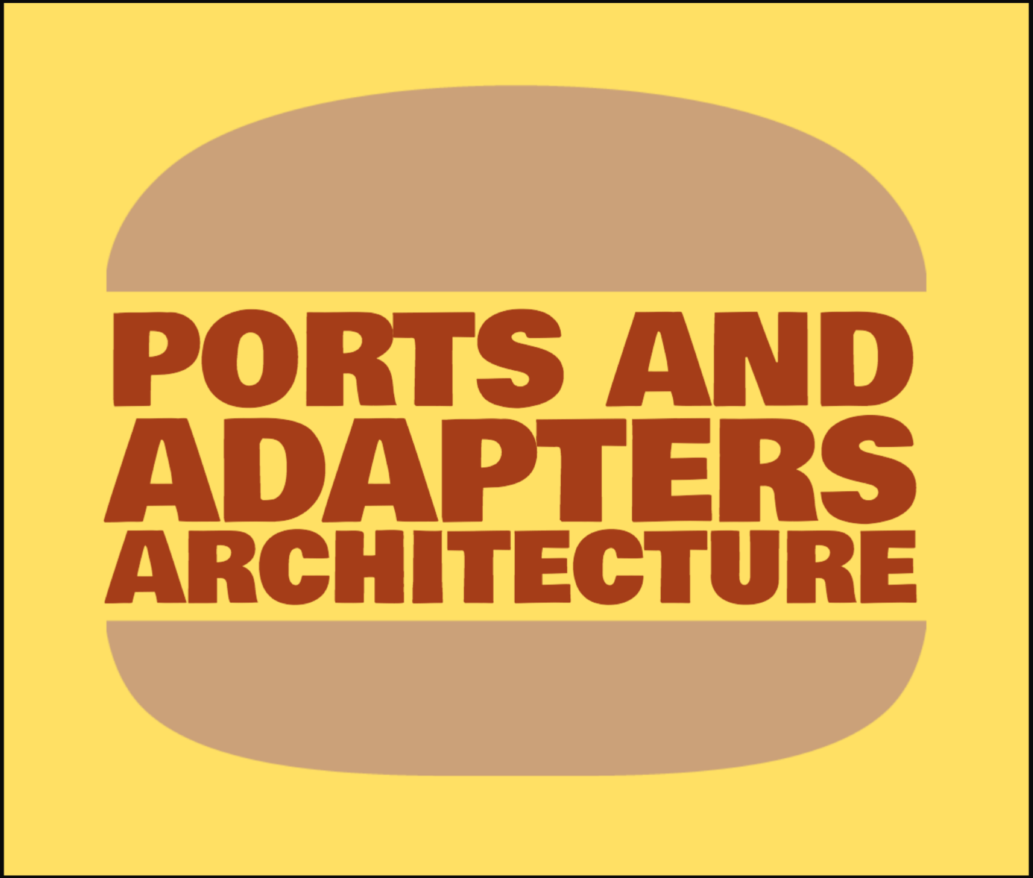
Bringing it all Together

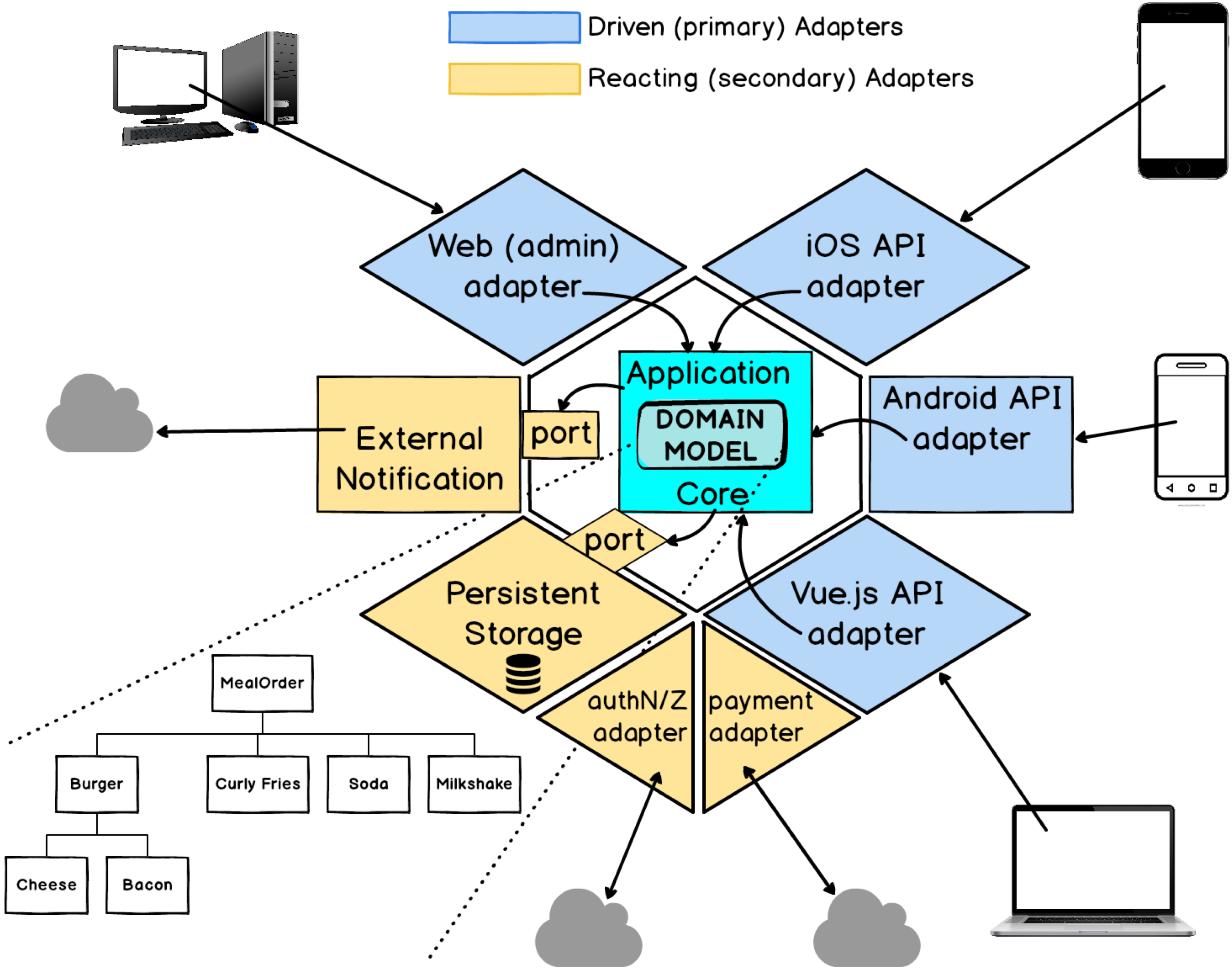
Ted M. Young

Java/Spring Trainer
& Technical Coach

Spiral Learning, LLC
ted@SpiralLearningLLC.com

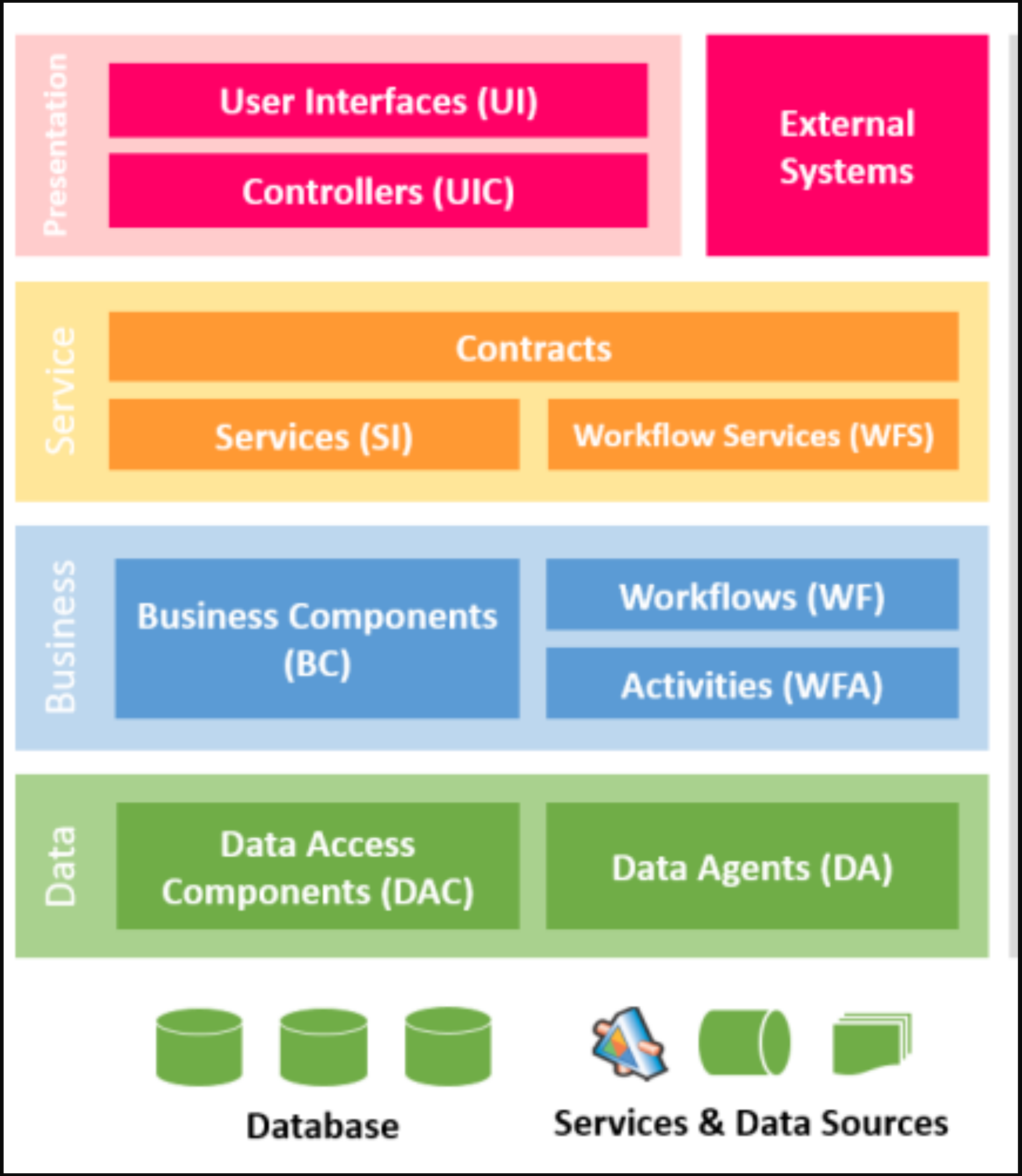


with Spring Boot, HTTP APIs, and more


Driven (primary) Adapters
Reacting (secondary) Adapters
Web (admin) adapter
iOS API adapter
Application
DOMAIN MODEL
Core
port
External Notification
Android API adapter
port
Persistent Storage
Vue.js API adapter
authN/Z adapter
payment adapter
MealOrder
Burger
Curly Fries
Soda
Milkshake
Cheese
Bacon

Layered Architecture

aka
Cake Architecture



Adapted from https://blogs.msdn.microsoft.com/malaysia/2013/08/02/layered-architecture-for-net/

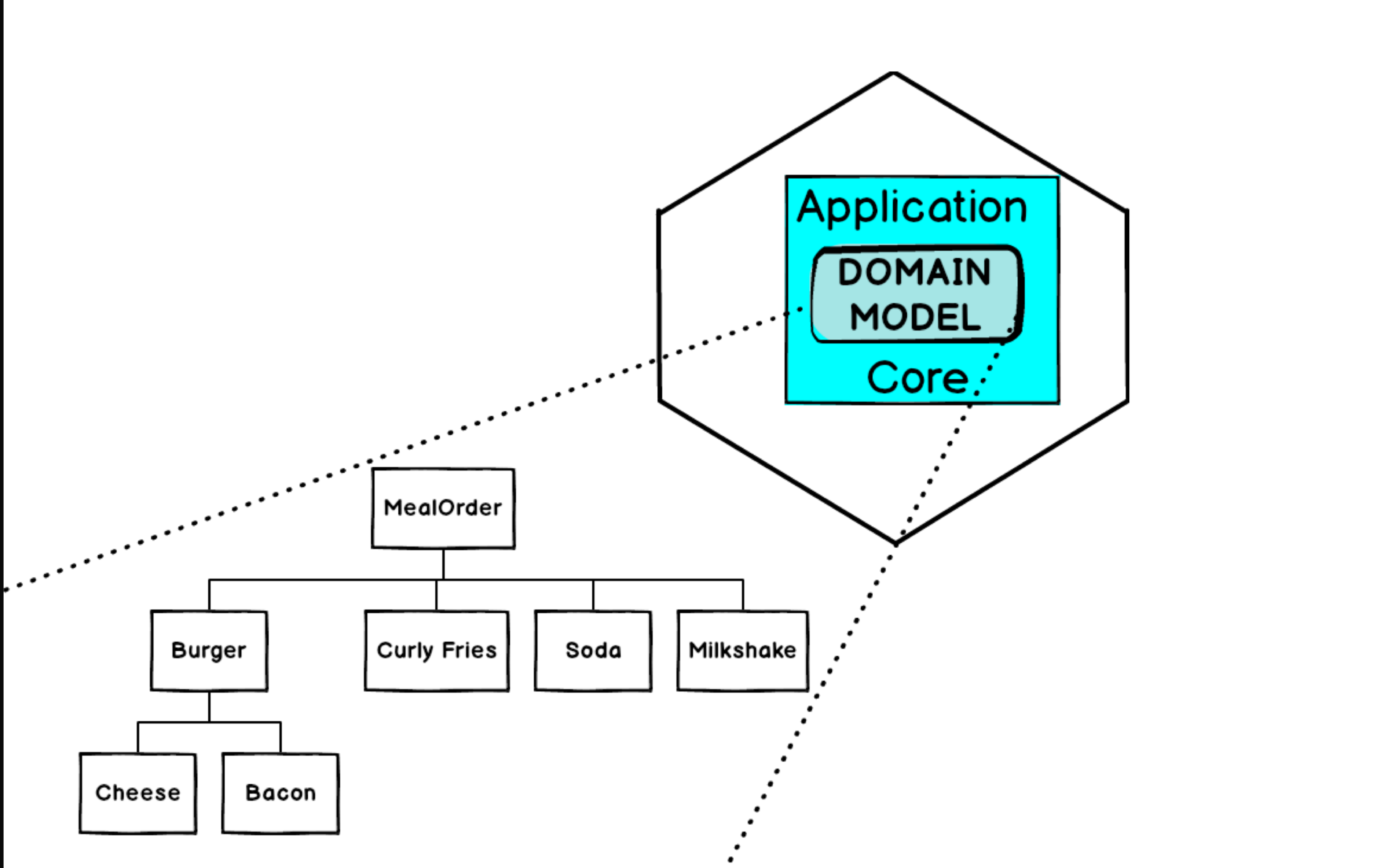
Database-Centric

instead of Business-Centric

Domain-Driven

Domain is focus of the application

Domain is Core



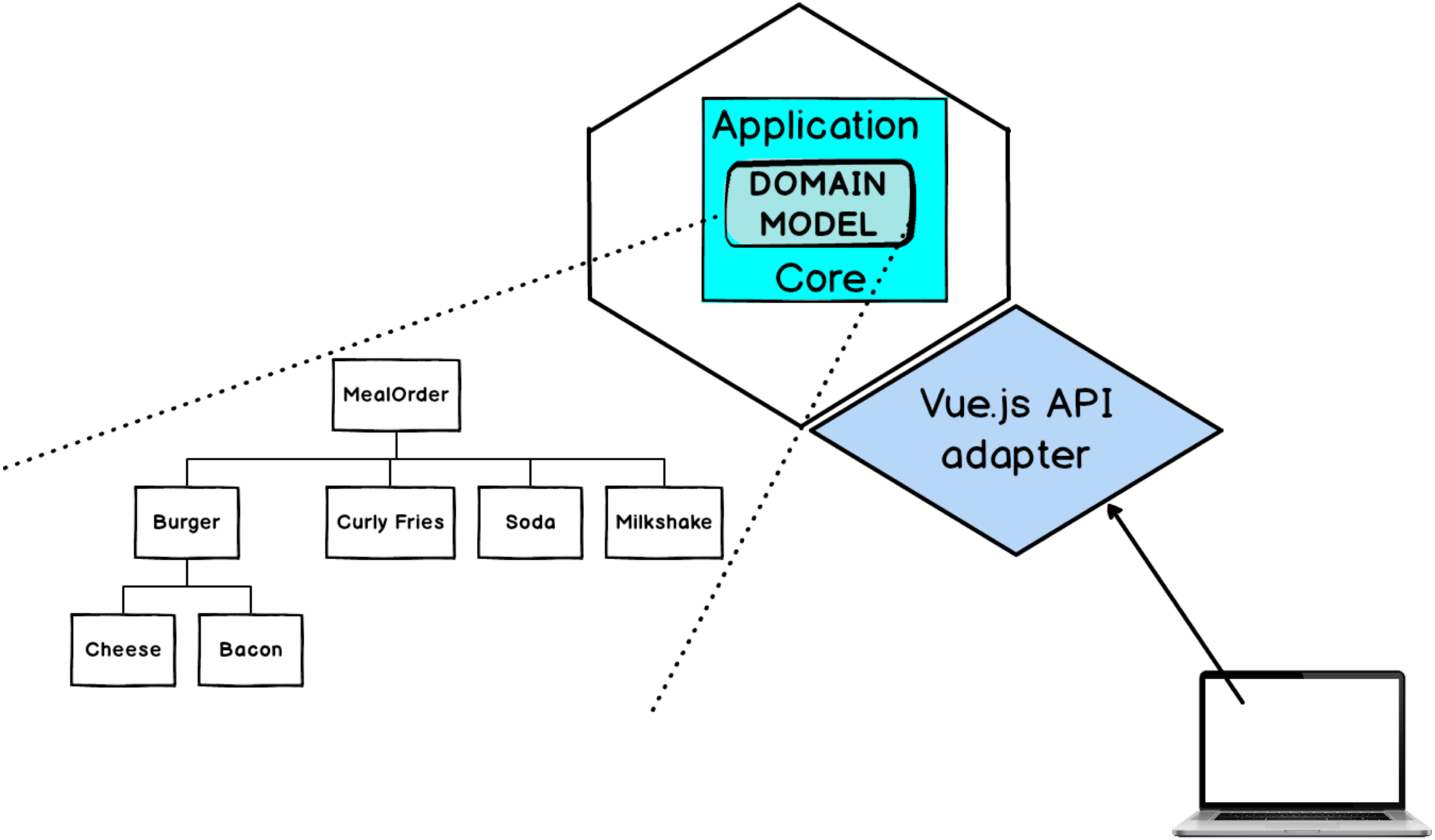
Ports & Adapters

Defines Crossing: Outside->Inside

Adapters = API

Define & Support External Driver

🟦 Driven (primary) Adapters



Application

DOMAIN MODEL

Core

Vue.js API adapter

MealOrder

Burger

Curly Fries

Soda

Milkshake

Cheese

Bacon

JavaBeans & DTOs

- ➢Adapters

vs. POJOs

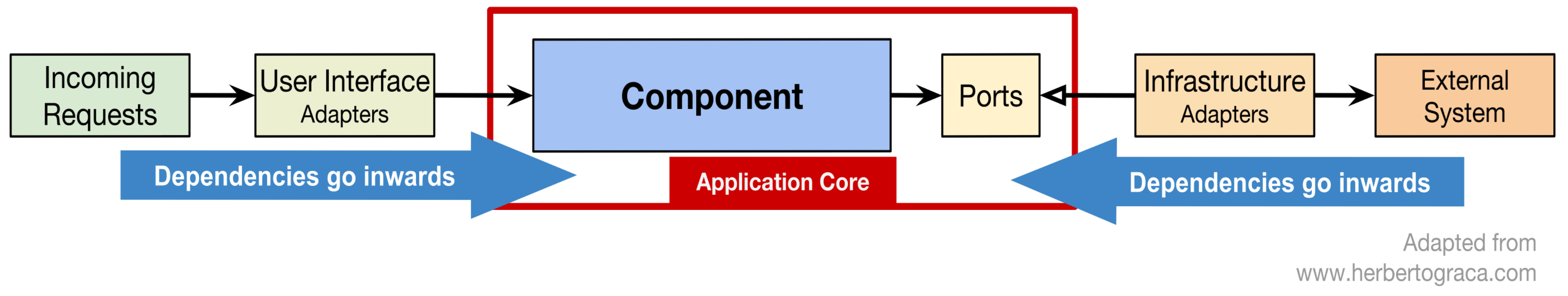
- ➢Domain

Adapters

- Repository
  - inside the domain
- Implementations
  - database
  - LDAP or user auth systems like Okta

Dependency Rules

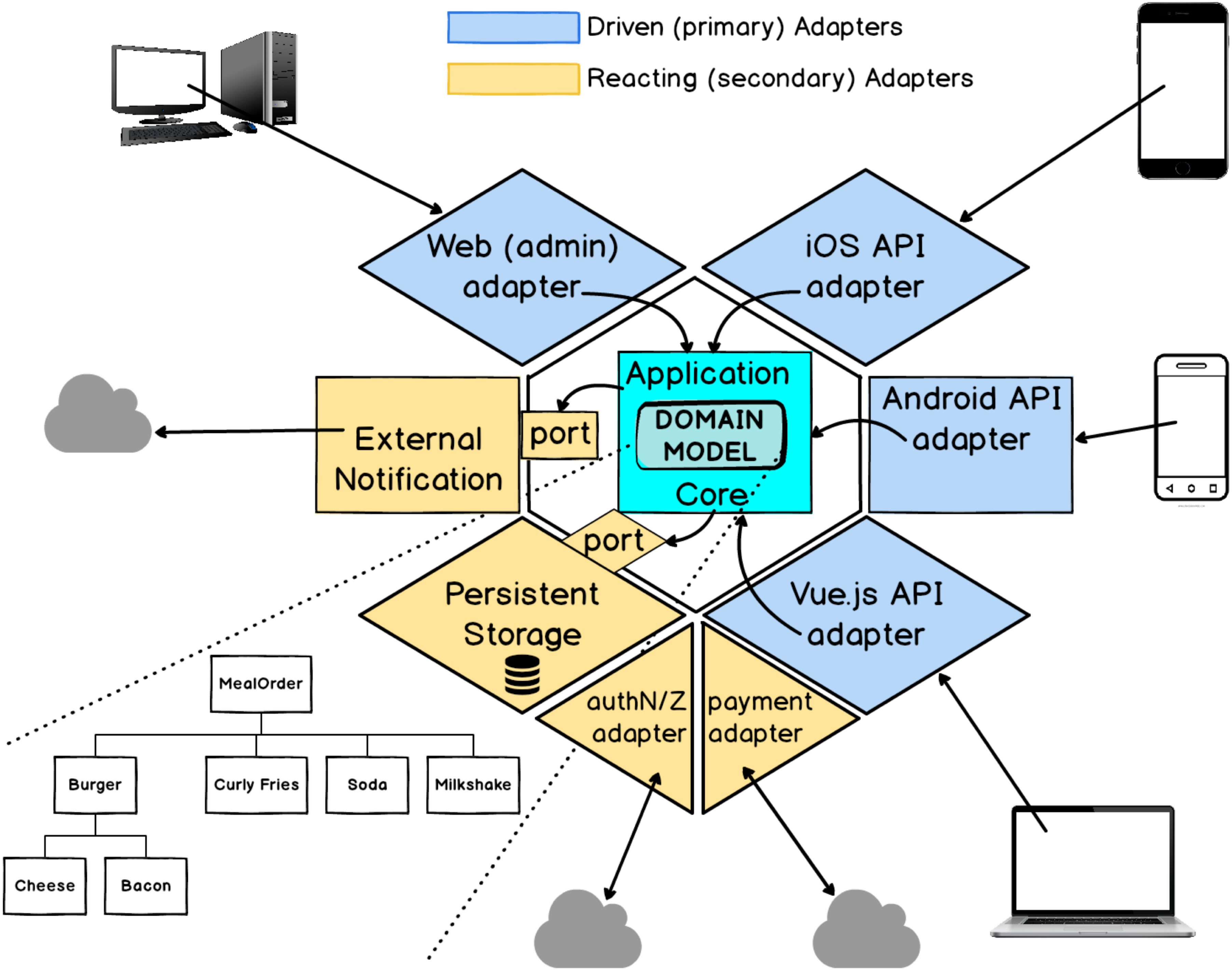
Dependencies point from outside to inside



Adapted from
www.herbertograca.com

Dependency Rules

- Dependencies point from outside to inside
- Adapters don't depend directly on other adapters
- Outgoing/domain-initiated flow relies on Dependency Inversion


Driven (primary) Adapters
Reacting (secondary) Adapters
Web (admin) adapter
iOS API adapter
Application
DOMAIN MODEL
Core
port
External Notification
Android API adapter
port
Persistent Storage
Vue.js API adapter
authN/Z adapter
payment adapter
MealOrder
Burger
Curly Fries
Soda
Milkshake
Cheese
Bacon