

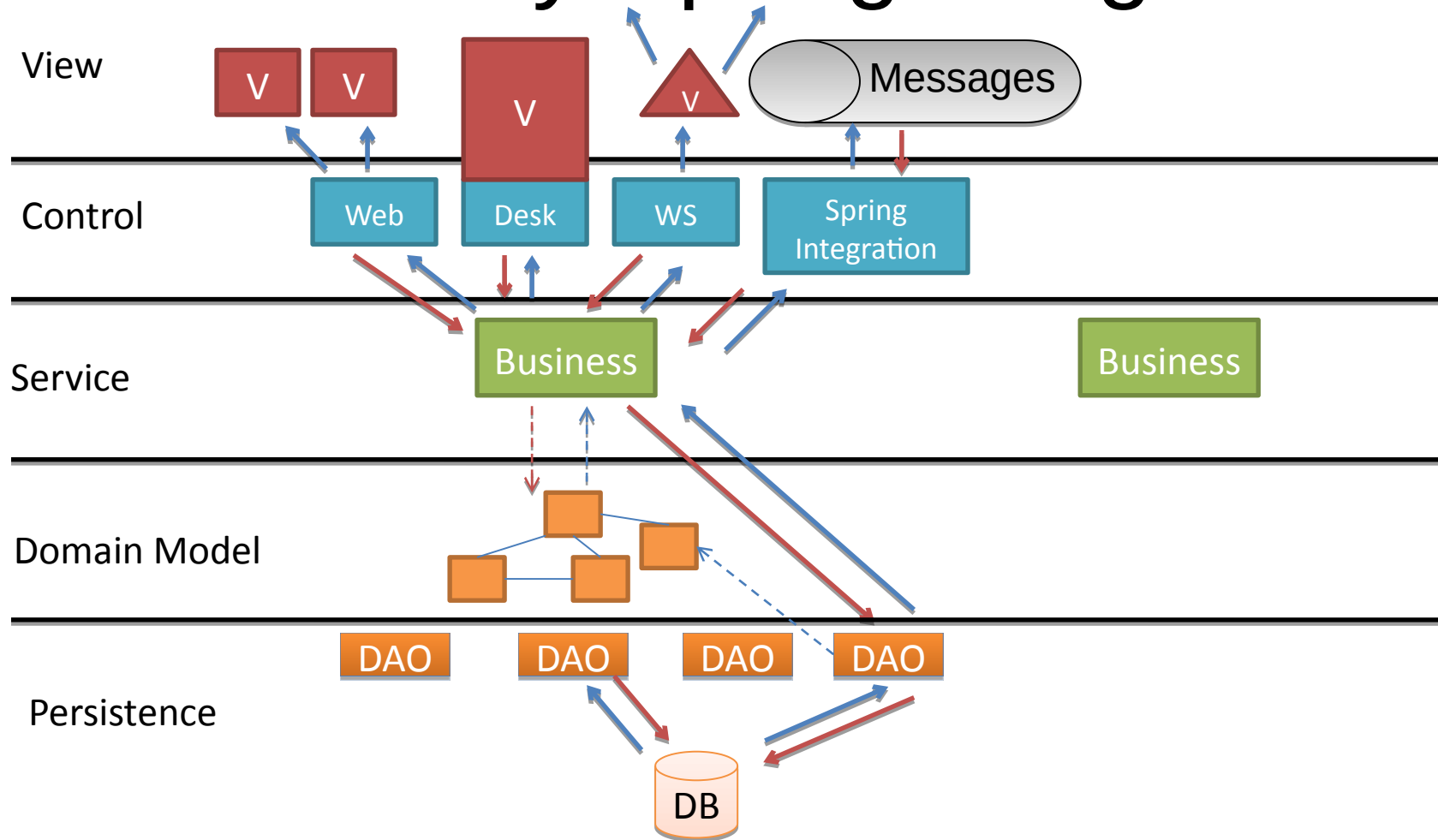


CS544 EA
Integration
Integration

Spring Integration

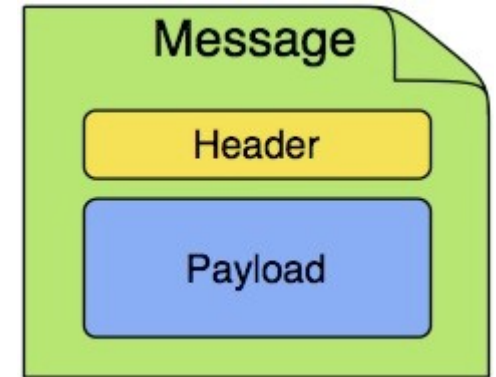
- The Spring Integration project is to Application Integration what Spring MVC is to HTTP.
 - **A thin layer interacting with service methods**
- But messages can come from through any of:
 - File system, shared DB, Remote Call, Messaging middleware
 - Can connect with all of these, and therefore is more abstract

Visually Spring Integration



Message

- In Spring Integration a message is a generic wrapper for **any Java object** combined with metadata (key/value pairs known as headers).
 - Any object can be a message
 - Spring Integration will add headers
 - Unlike the messaging middleware it can come from / go to anywhere

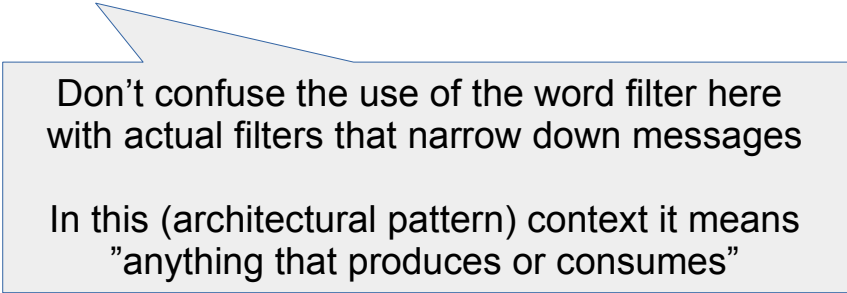


Decoupling

- The whole purpose of messaging is to keep systems **as decoupled as possible**, while still integrating.
 - Each system should be able to work regardless of the protocols, formatting, or implementation details of the messages and the sender / receiver of them.

Pipes-and-Filters

- The Spring Integration documentation calls:
 - **Pipes** transport the messages between filters so that components remain loosely coupled
 - **Filters** are any component capable of producing or consuming messages



Don't confuse the use of the word filter here with actual filters that narrow down messages

In this (architectural pattern) context it means "anything that produces or consumes"