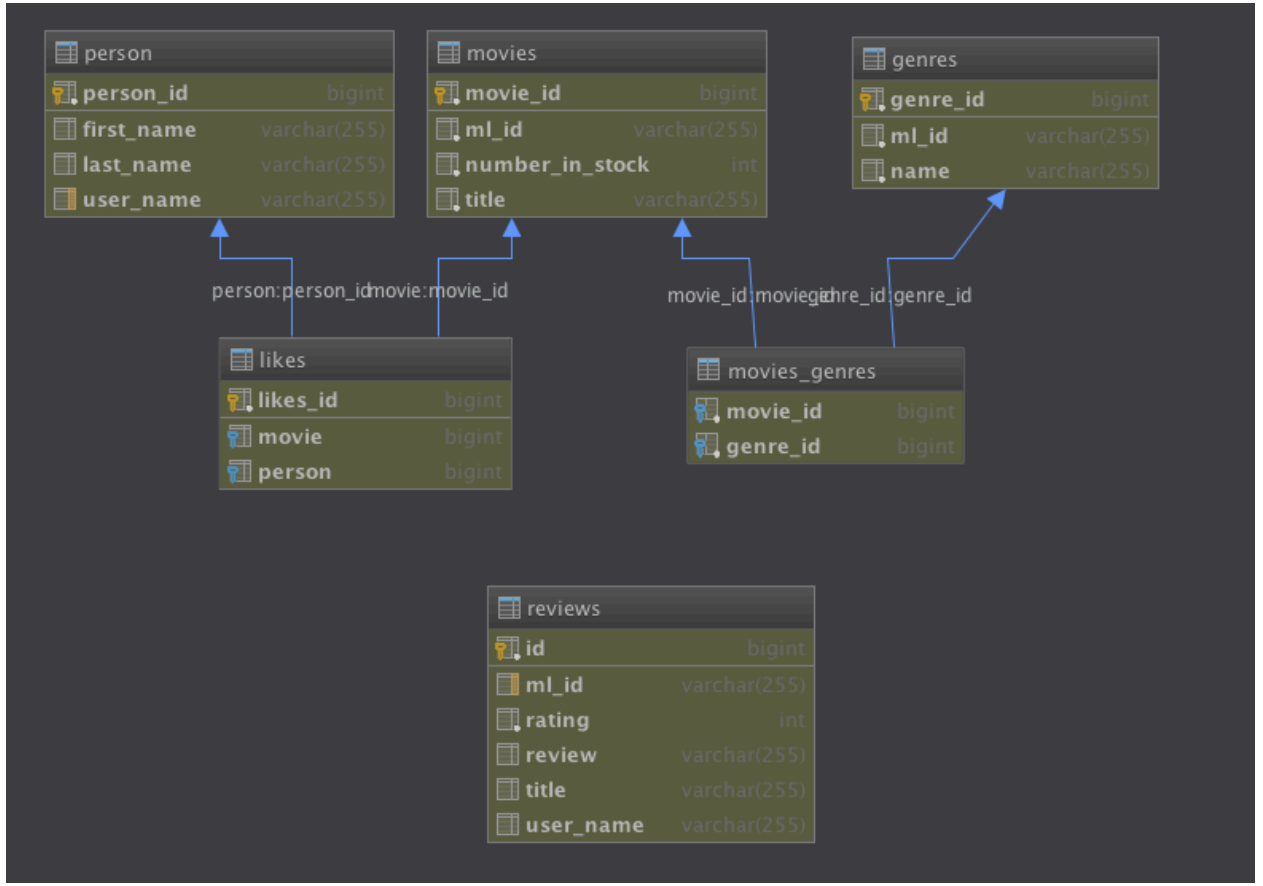
Splitting the Monolith

Session 04

Breaking SpringBox apart

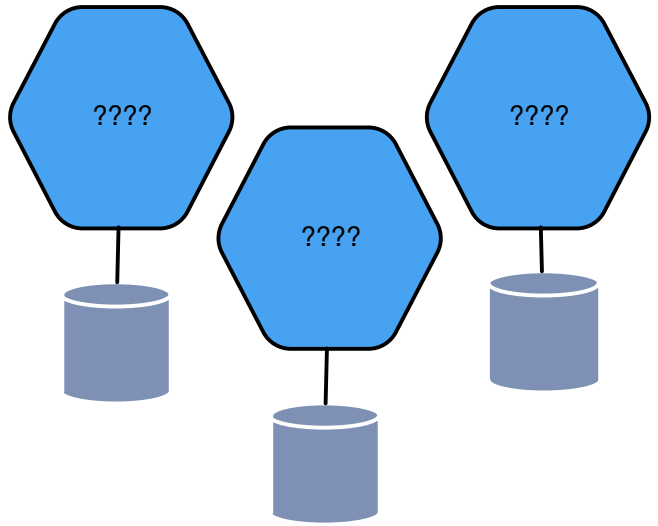
- Data domain
- Data decomposition



Pivotal

Polyglot persistence

- Decomposition allows polyglot persistence
  - Data stores are chosen based on data shape and read/write patterns.

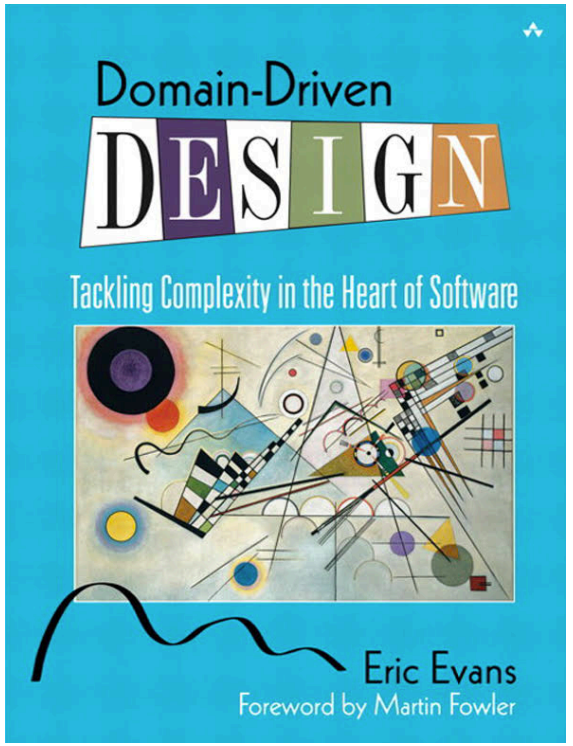


Decomposition recipes

- **DON'T DECOMPOSE THE MONOLITH!! (at least immediately)**
- Add new features as Microservices
  - How to make new features (microservices) talk to the monolith?

The Anti-corruption Layer

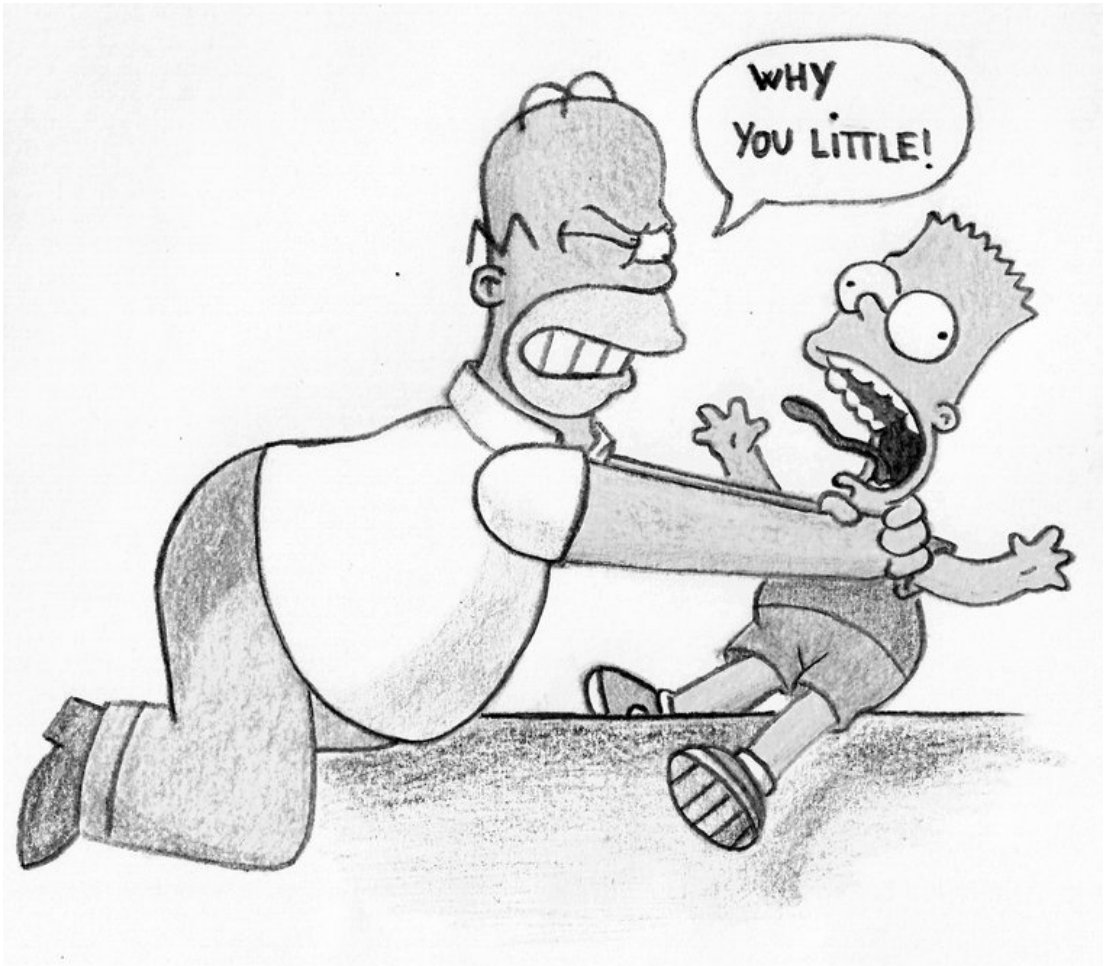
- Allows the integration of two systems without domain models corrupt each other.
- Essentially an API contract
- Based on 3 sub-modules:
  - Façade – system integration
  - Adapter – protocol translation
  - Translator – model translation



Pivotal

Strangling the Monolith

- Identify bounded contexts.
- Prioritize – each ones will benefit most?



Pivotal

Remedies when moving into Cloud

Anti-Cloud Pattern	Remedy
Multiple Inbound protocols like RMI, JMX, Custom-TCP	TCP Routing, Tunneling
Persistence in local VMs like Caches, Transaction logs, CMS	External managed data services, Persistent volume support, sshfs service, EBS & S3 blob store, mount external NFSTransations
JTA & 2-pc commit	Use standalone transaction managers like Atomikos and Bitronix. Introduce eventual consistency patterns
Spaghetti Configuration	Externalize, plugin config via Config Server , init script in .profile.d directory that injects the config. as environment variables

http://cloud.rohitkelapure.com/2015/08/porting-strategies-for-migrating-apps.html

Pivotal

Labs!!