More Object Design with GoF Patterns

Curt Clifton

Rose-Hulman Institute of Technology

Applying Patterns to NextGen POS Iteration 3

- Local caching
- Failover to local services
- Support for third-party POS devices
- Handling payments

Applying Patterns to NextGen POS Iteration 3

- Local caching
- Failover to local services
- Support for third-party POS devices
- Handling payments

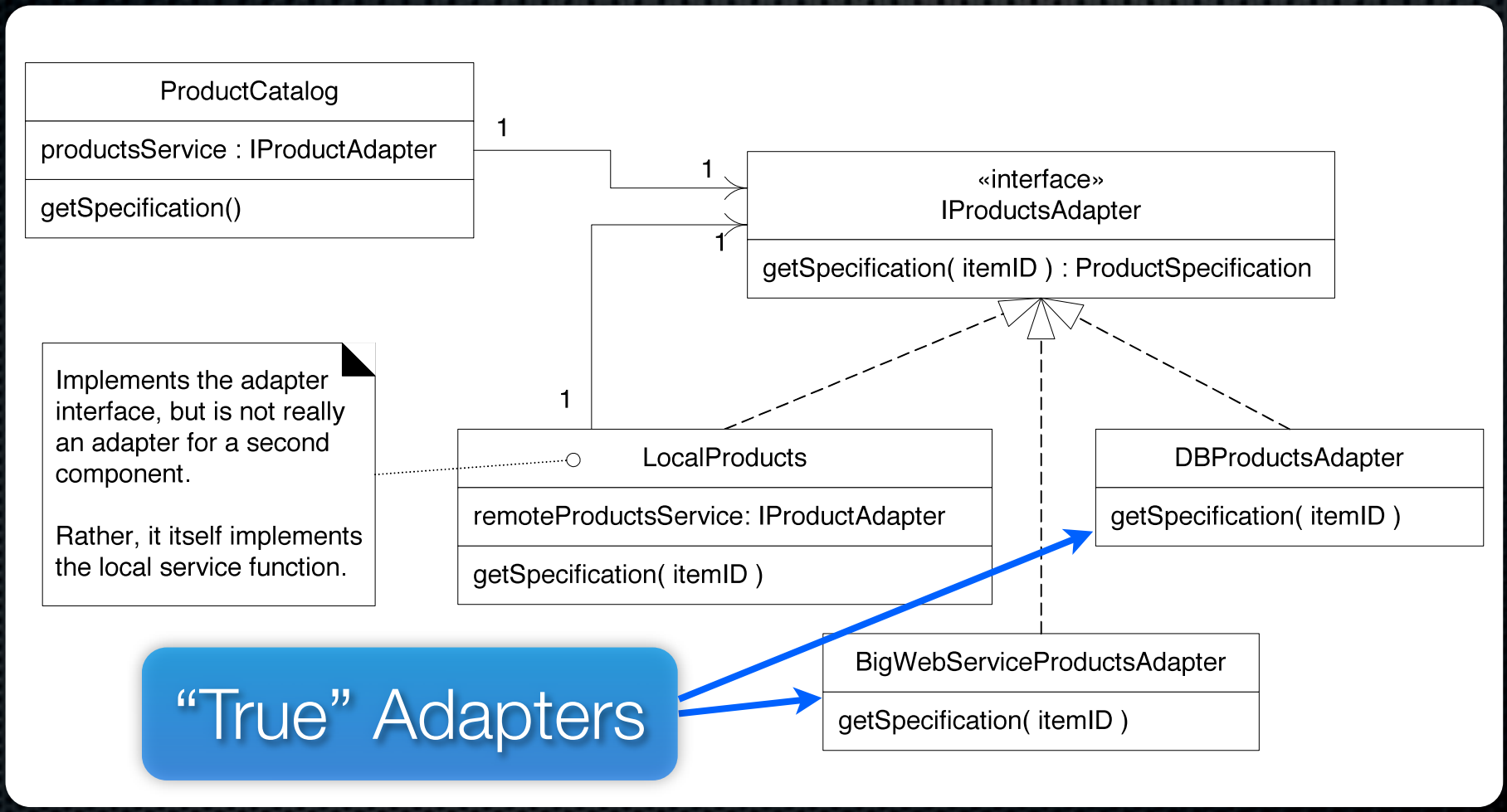
Failover and Performance with Local Caching

- What is a *cache*? How does a cache usually work?
- Why use a local cache for NextGen POS?
  - Performance
  - Improve recoverability

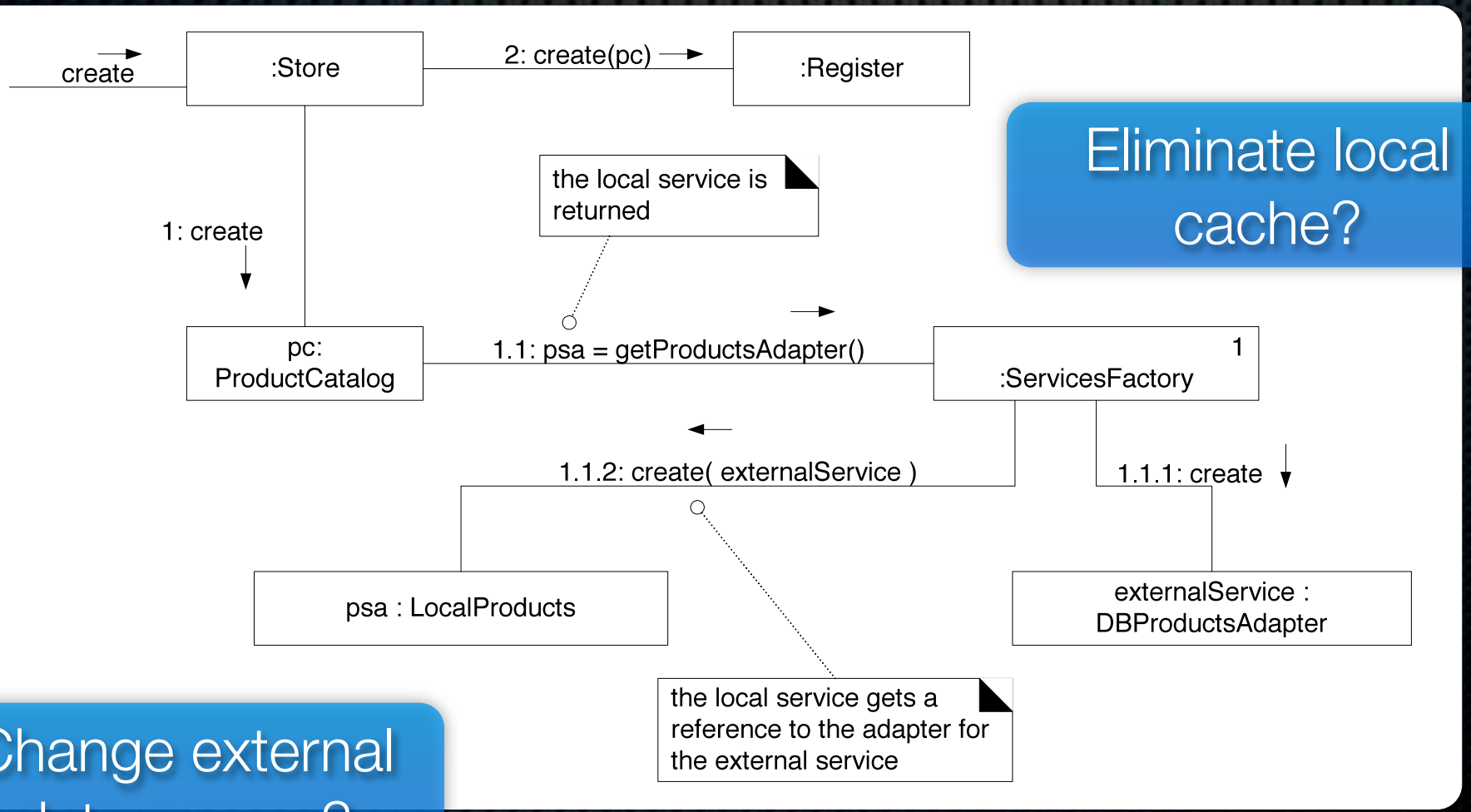
Search Strategy for Product Information

1. Look in memory (in map stored by *ProductCatalog*)
2. Look on local hard drive cache
3. Retrieve from remote persistence service

Applying the Adapter Pattern



Using a Factory to Set Up for Local Caching

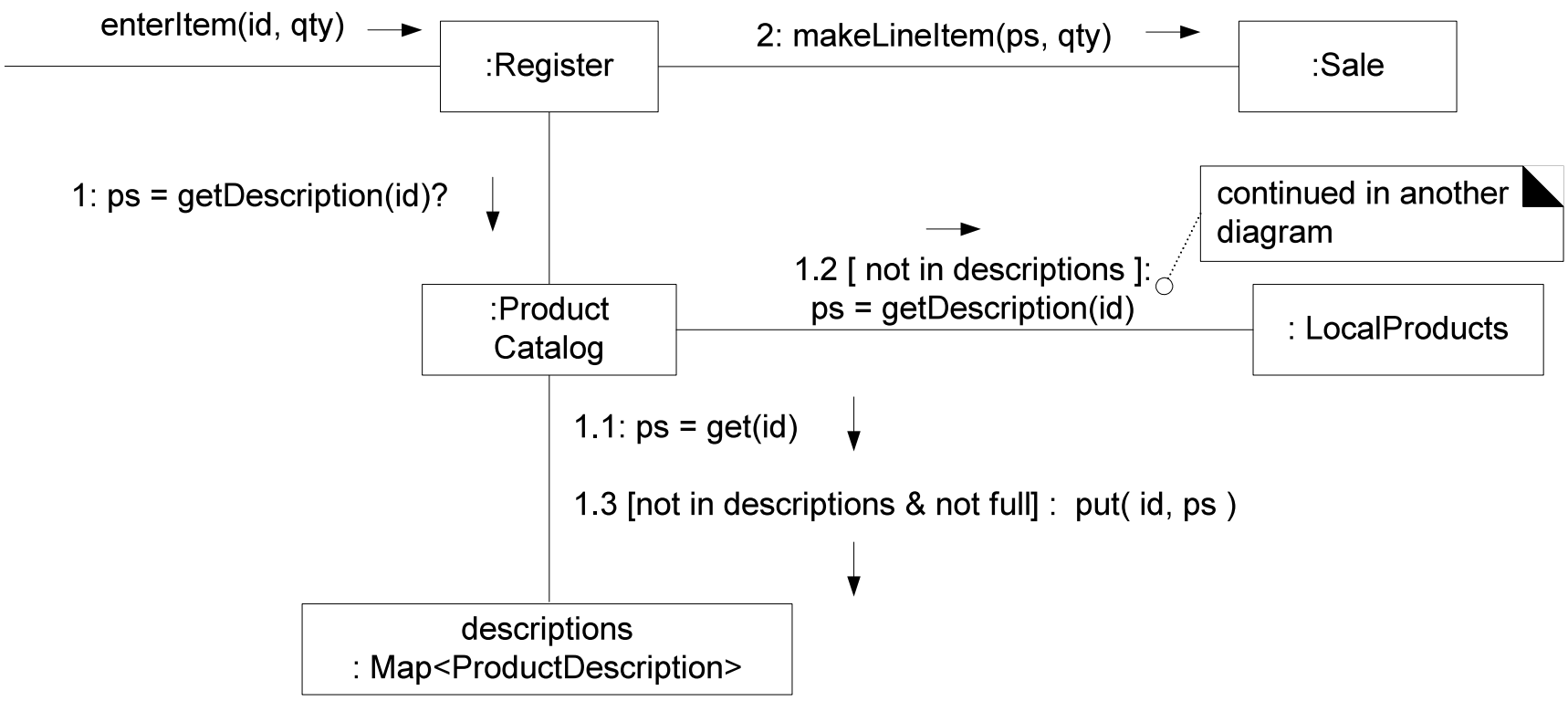


Eliminate local cache?

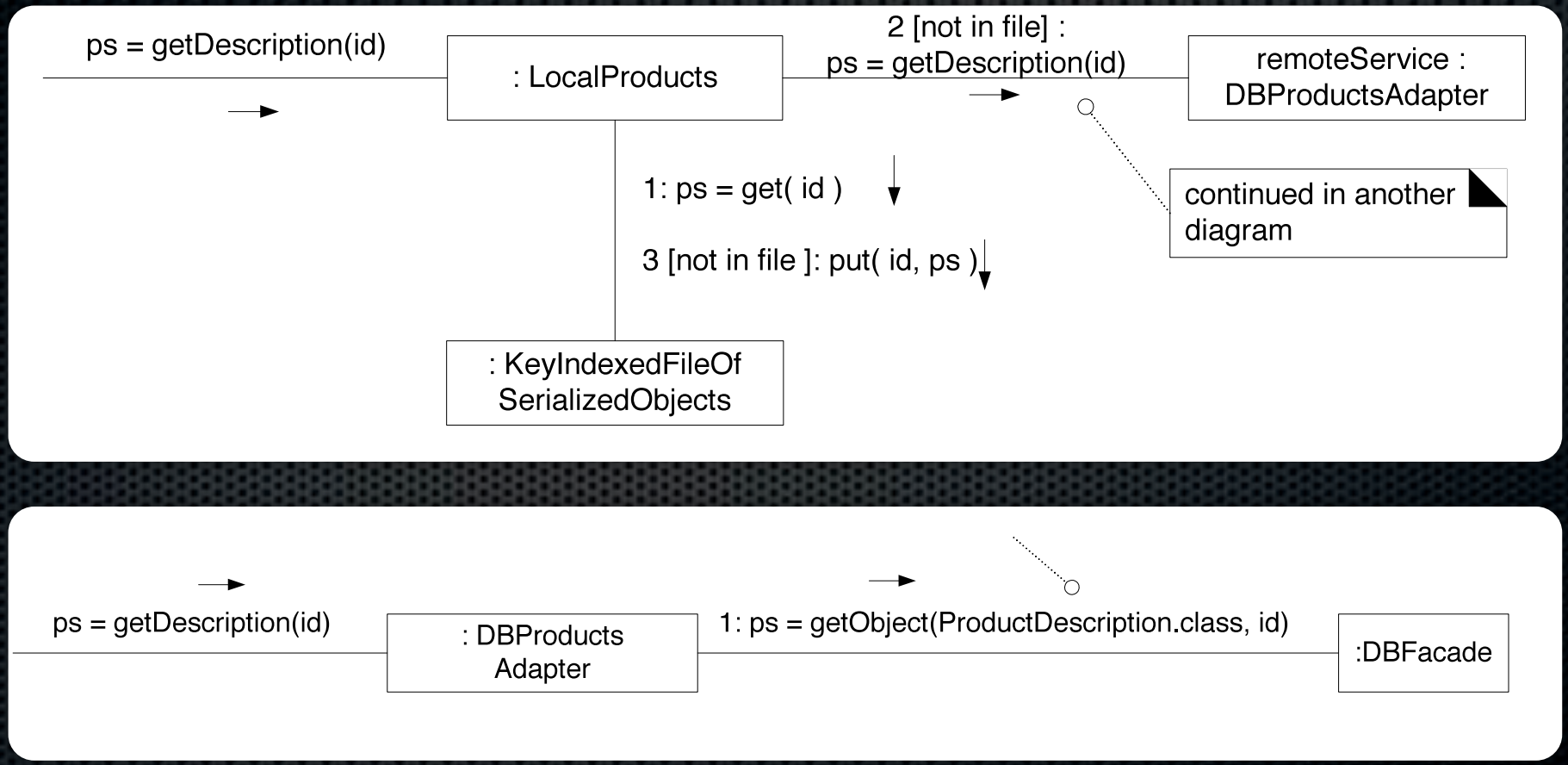
Change external data source?

Q2

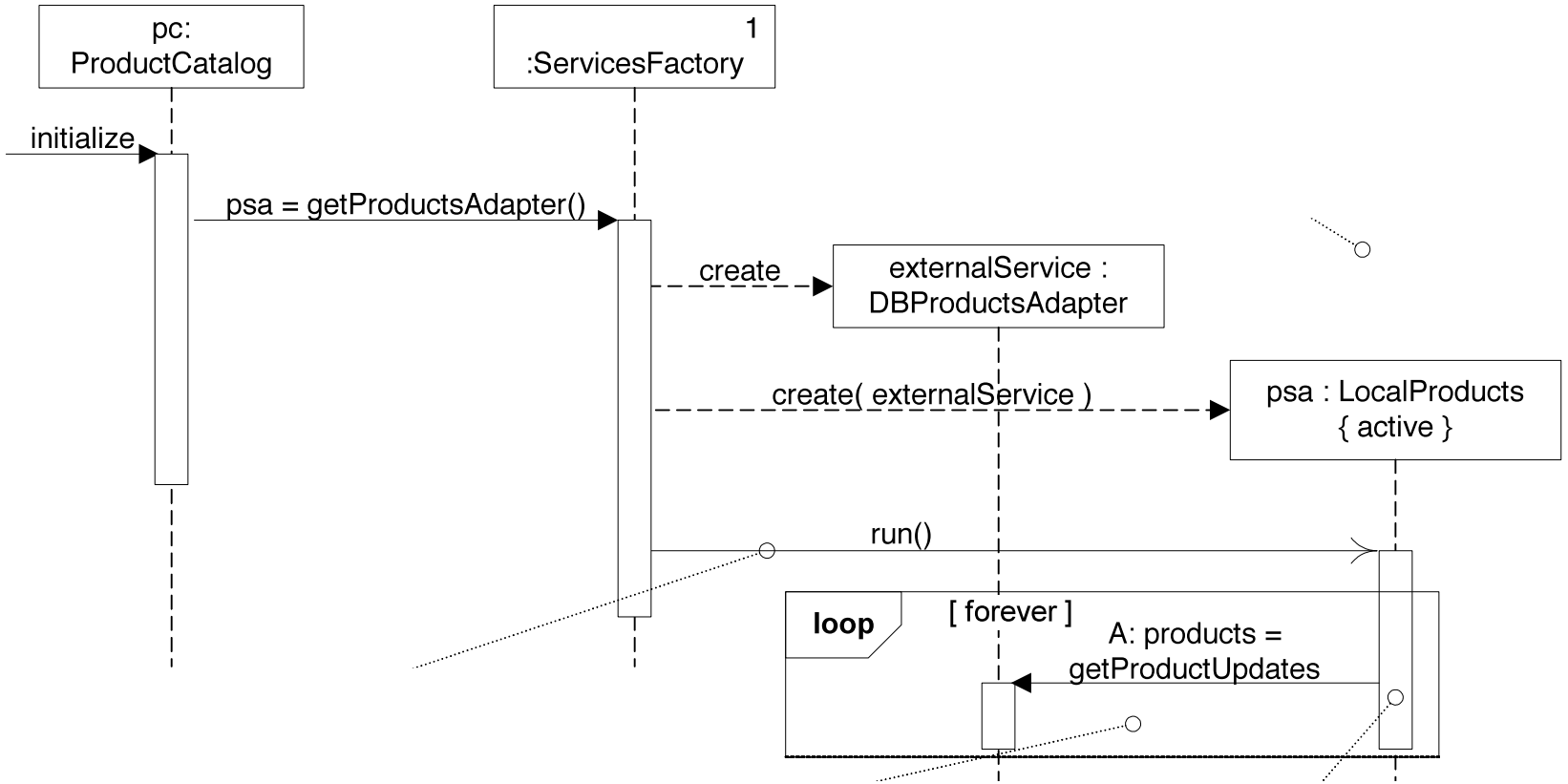
Product Lookup with In-memory Miss



Product Lookup with Local Cache Miss



Using Threads to Freshen the Cache



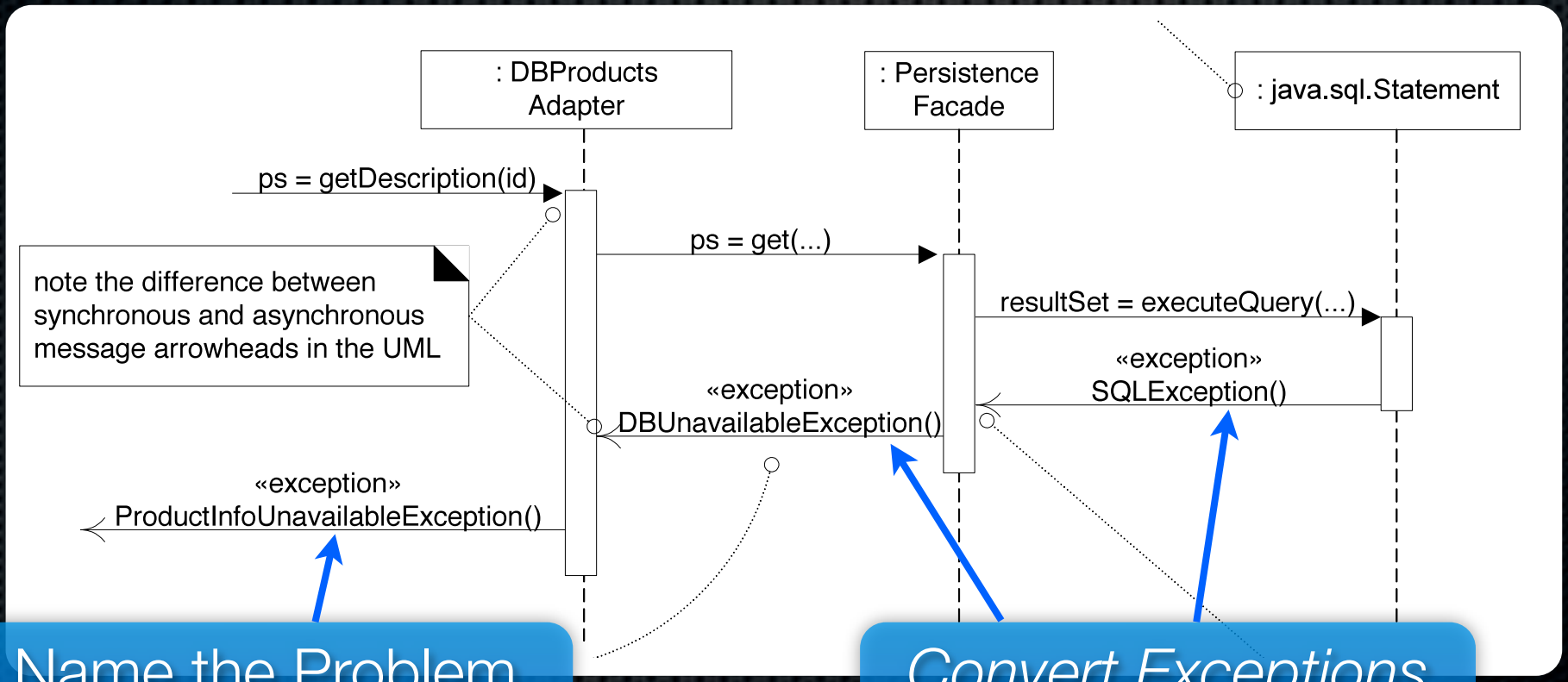
How's the final iteration going?

Used by permission. http://notinventedhe.re/on/2009-12-28

Handling Failure in NextGen POS

- What should happen if there is a local cache miss and the external product information service fails?

Showing Exception in Sequence Diagrams



Name the Problem not the Thrower

Convert Exceptions pattern

Q3

How should NextGen POS handle this exception?

- Common exception handling patterns
  - Use a central error logging object to record all exceptions for diagnosis by developers
  - Use a standard, application-independent, non-UI object to notify users
    - Can delegate to multiple different UI notifications
    - Protected Variation for changes in reporting

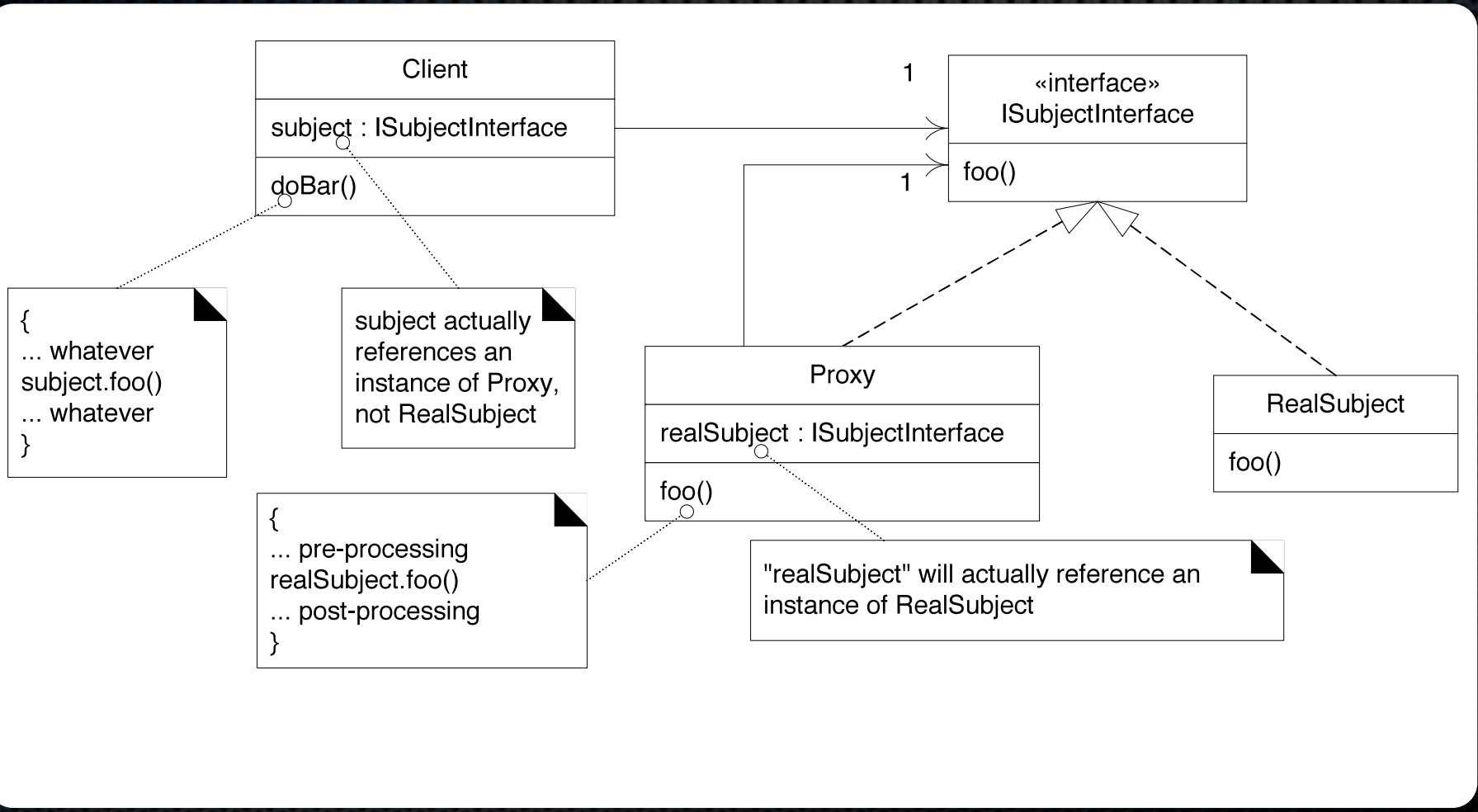
Failover to Local Services with a Proxy

Proxy GoF Pattern

- **Problem**: How do we control access to some *subject* object if we want to avoid giving direct access?
- **Solution**: Add a level of indirection with a *proxy* object that implements the same methods as the *subject* and conditionally delegates to it.

Q4,5

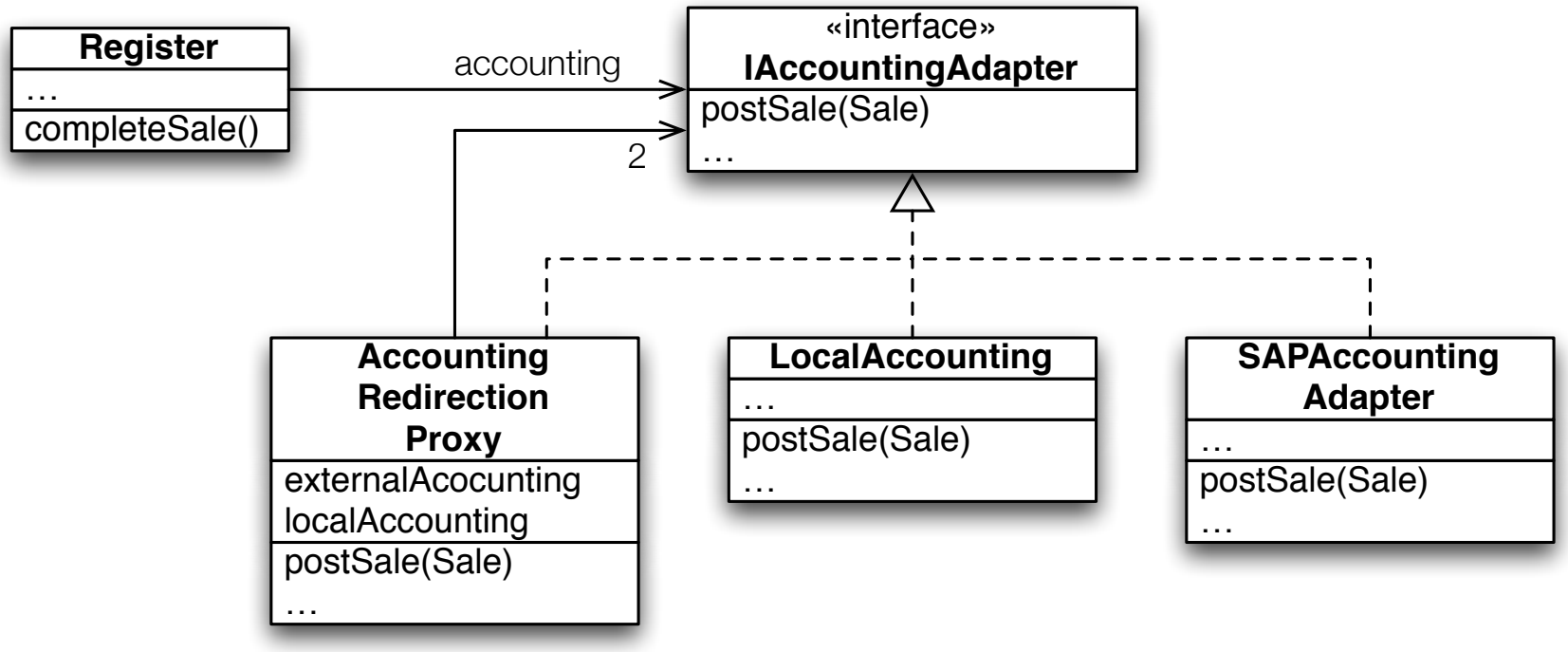
Structure of the Proxy Pattern



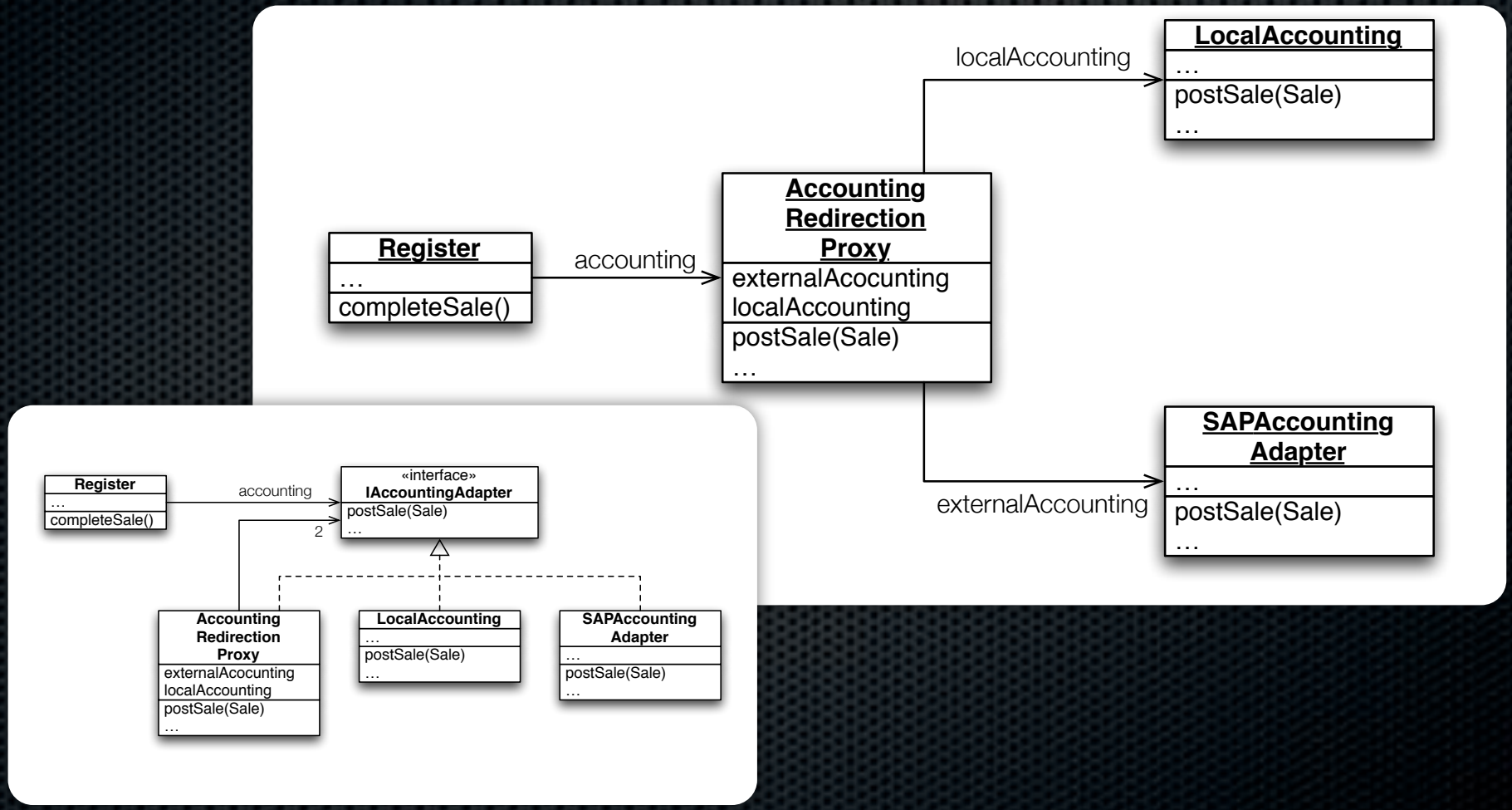
Proxy in NextGen POS

- Posting sales to the accounting service
  - Send *postSale(Sale)* to a *redirection proxy*
  - *Proxy* attempts to post to external service
    - If it fails, then *proxy* stores result locally

Proxy in NextGen POS—Class Diagram



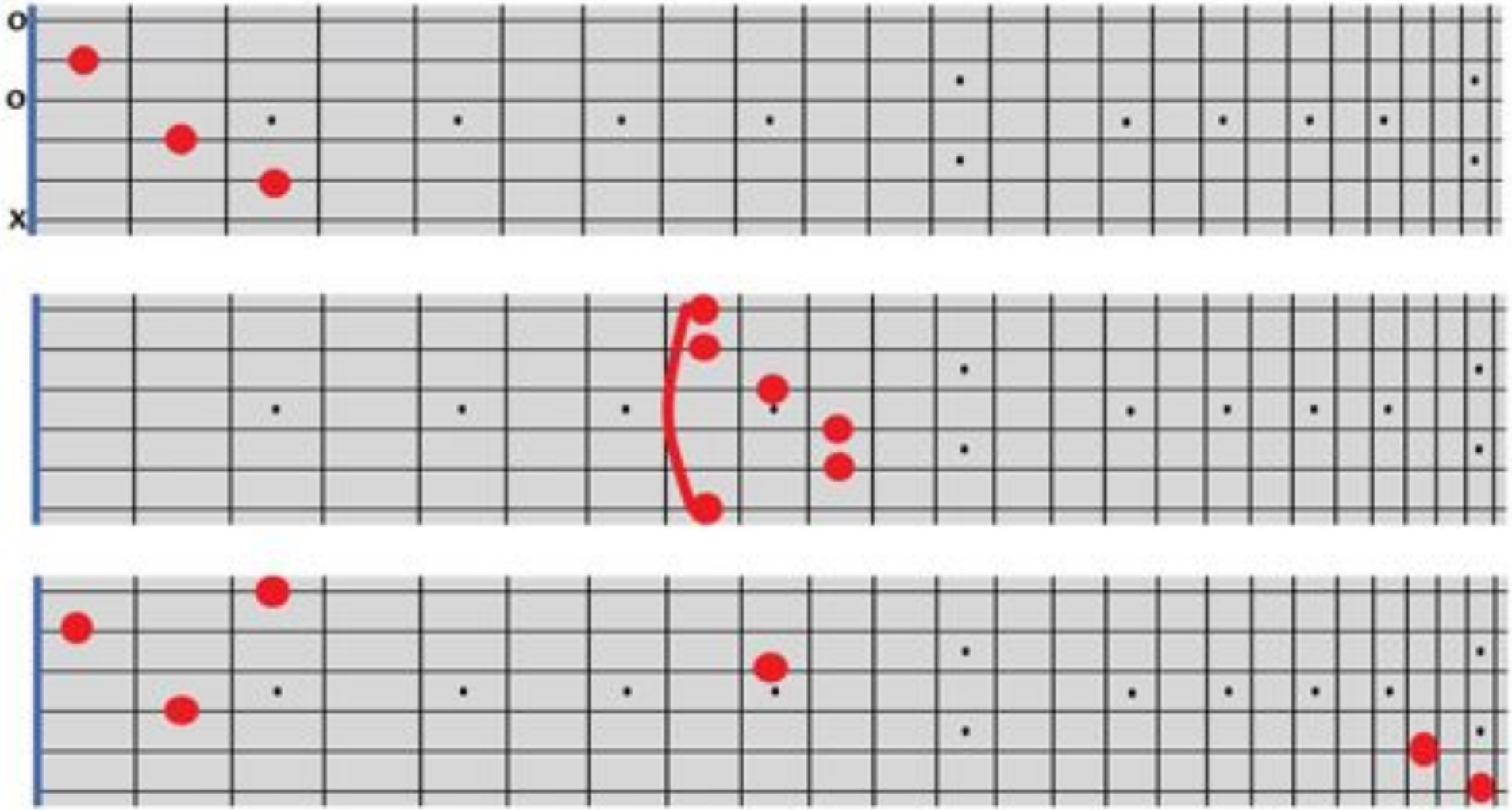
Proxy in NextGen POS— Object Diagram



Q6

Design Studio: Guitar Chord Finder

Team describes problem and perhaps current solution (if any)	~5 min.
Class thinks about questions, alternative approaches. **Q7**	~3 min.
On-board design	~12 min.



All valid C Major chords, however the last one is not playable