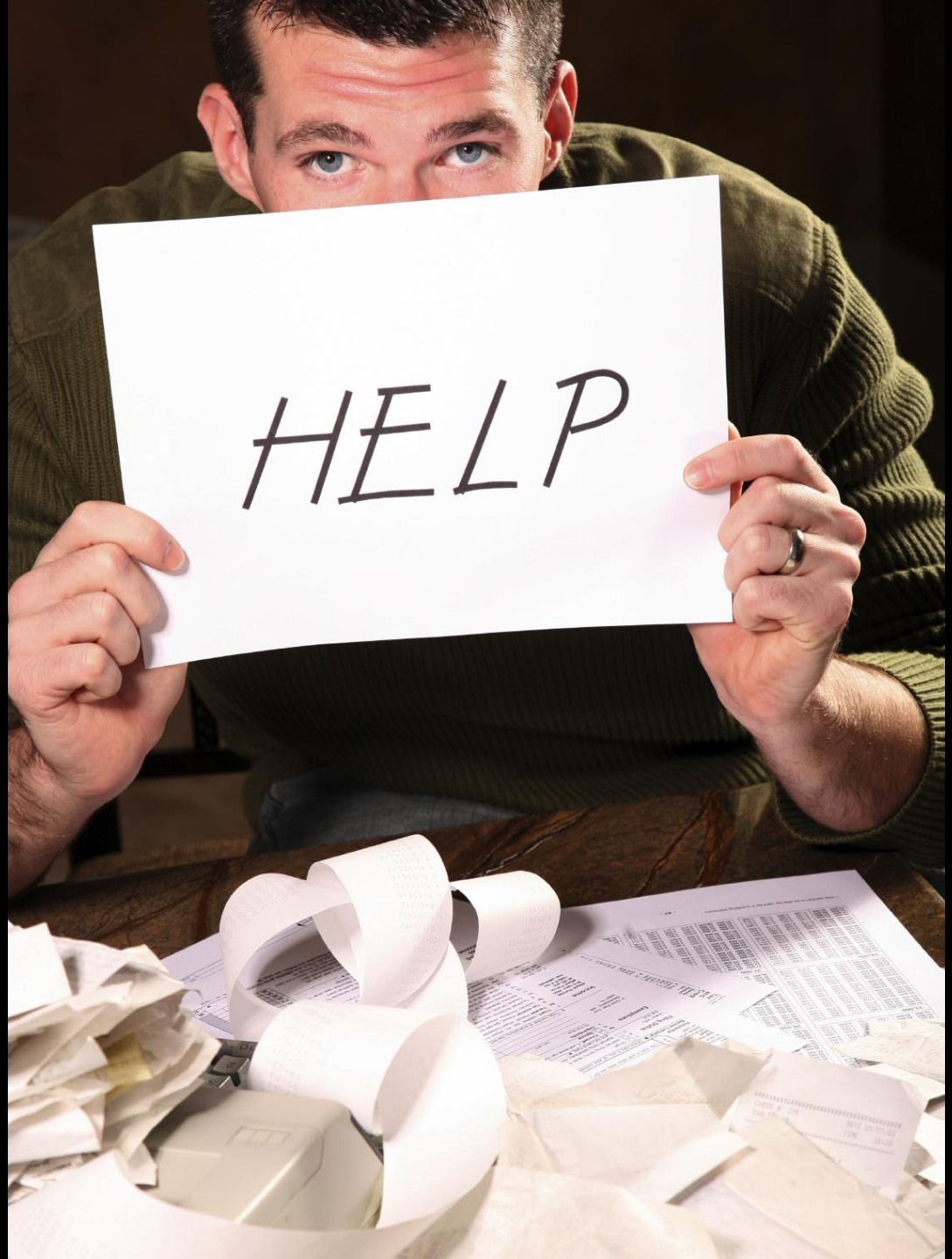
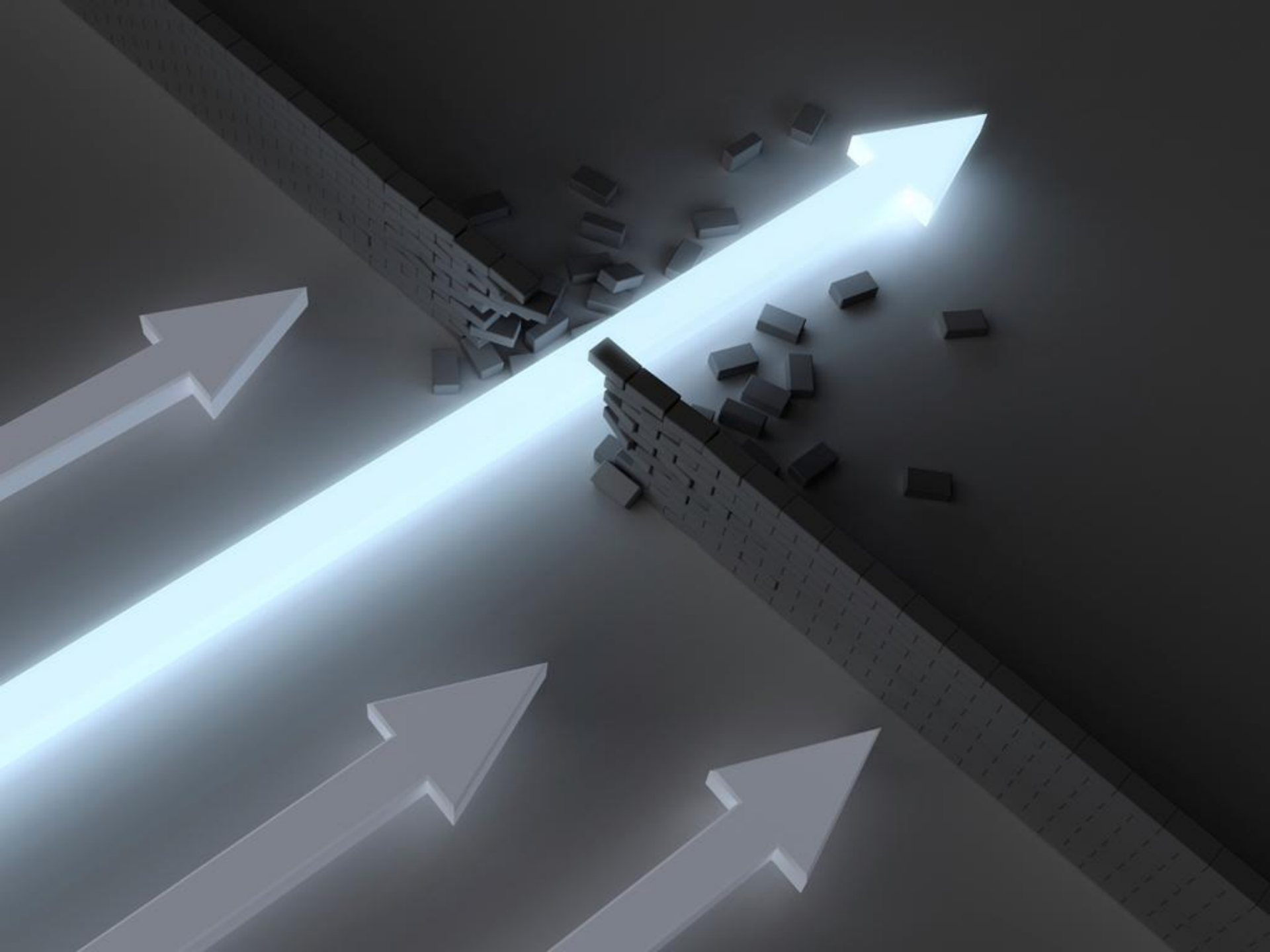

Unleash Your Domain
Greg Young

Agenda

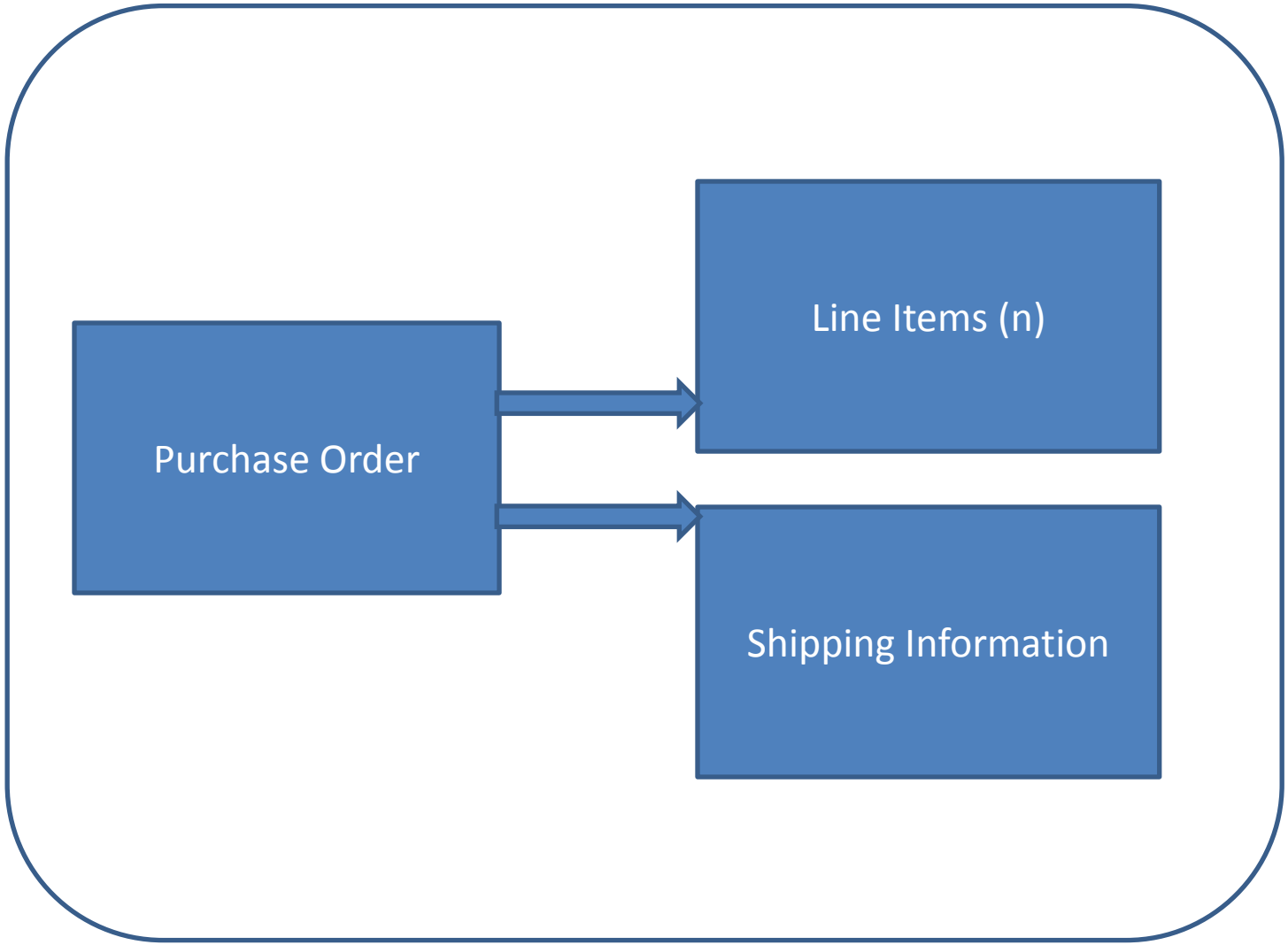
- The Issues
- The Breakthroughs
  - Explicit State Representation
  - Event Storage
  - Command Query Separation
  - Asynchronous Context Mapping
- Summary
- Questions

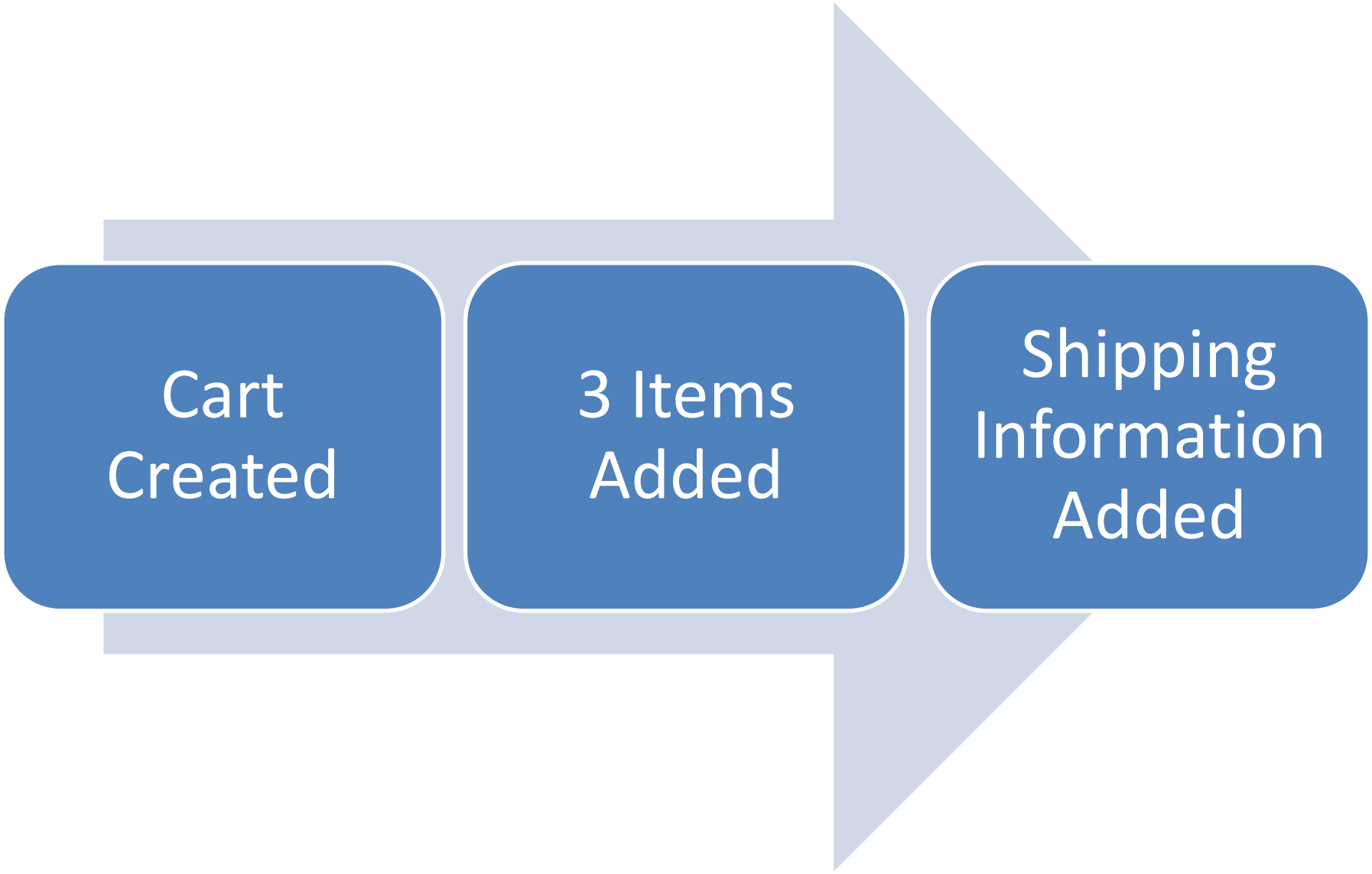

TOXICITY REPORTING FORM (continued)
Patient reference no .....
O/WNL
1 (Mild)
2 (Moderate)
3 (Severe)
4 (Unacceptable)
1. Stomatitis
2. Abdominal pain: severity treatment
3. Constipation
Diarrhoea
none
none
---
no chg
none
mild ileus
?2-3 stools/day
none
none
reasonable intake 1 x/day
>90
normal
80-89
tachypnea
>30
24-30
asymptomatic/ ?ej. Fr. <20%
asympt./transient increase by >20mmHg. no RX req
20-24
asymptomatic/ej. fr. <80% baseline
recur./persist increase by 20mmHg. no Rx req
<20
mild CHF responds to Rx
requires therapy
severe or refractory CHF
hypertensive crisis
2. Central: Cerebellar
no chg
no chg
mod sensory loss, mod paresthesias mild obj weakness / no sig impair
interferes with function
painful/odema can eat
moderate required – helps
mod ileus
?4-6 stools/day or mod. cramps
decreased intake 2-5 x/day
65-79
dyspnea
50-64
O_2 required
cannot eat or drink
moderate-severe required-no help
severe ileus
?7-9 stools/day or severe cramps
no sig. intake 6-10 x/day
requires parenteral or enteral support

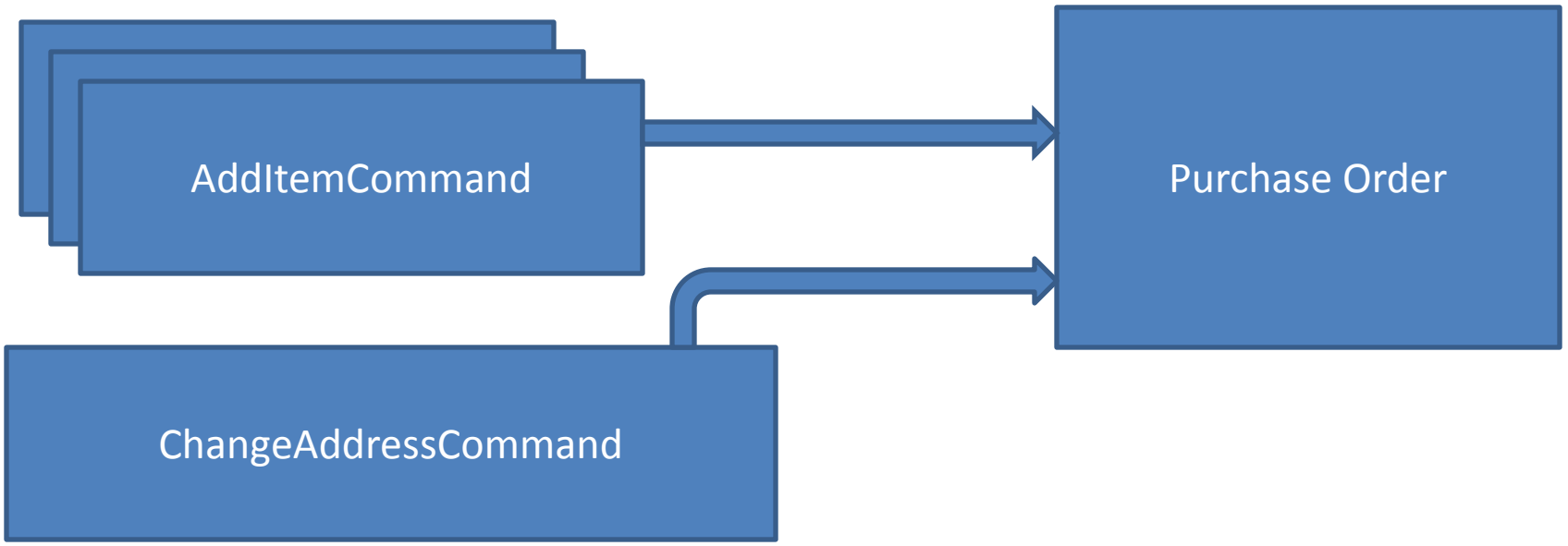

HELP



State transitions are an important part of our problem space and should be modeled within our domain.

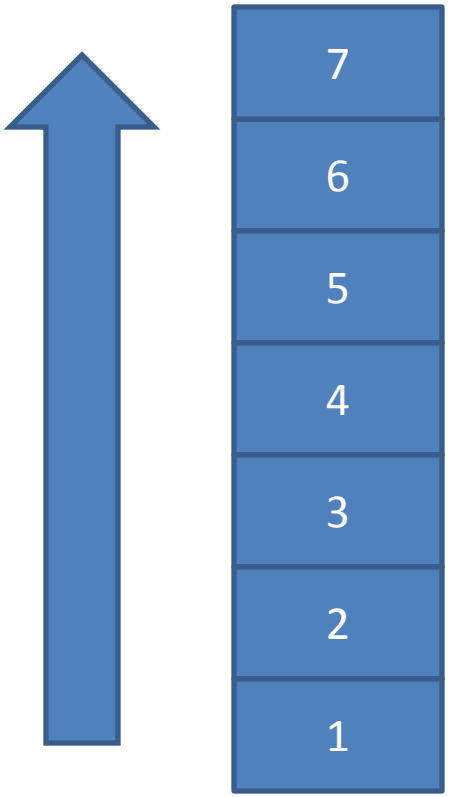

Purchase Order
Line Items (n)
Shipping Information

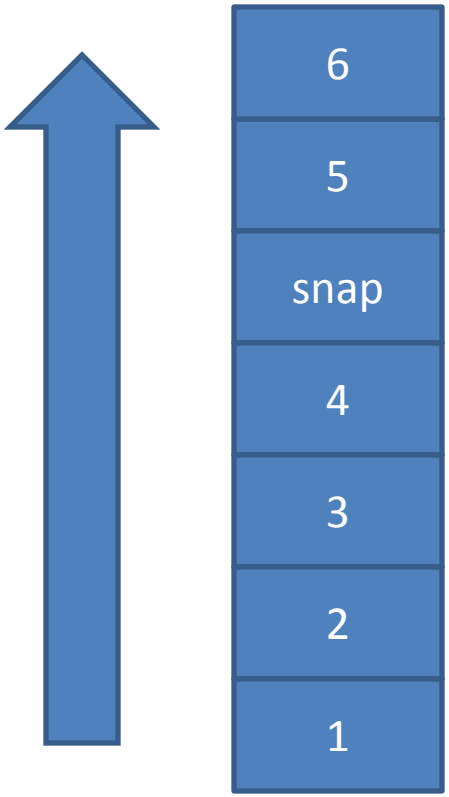

Cart Created
3 Items Added
Shipping Information Added

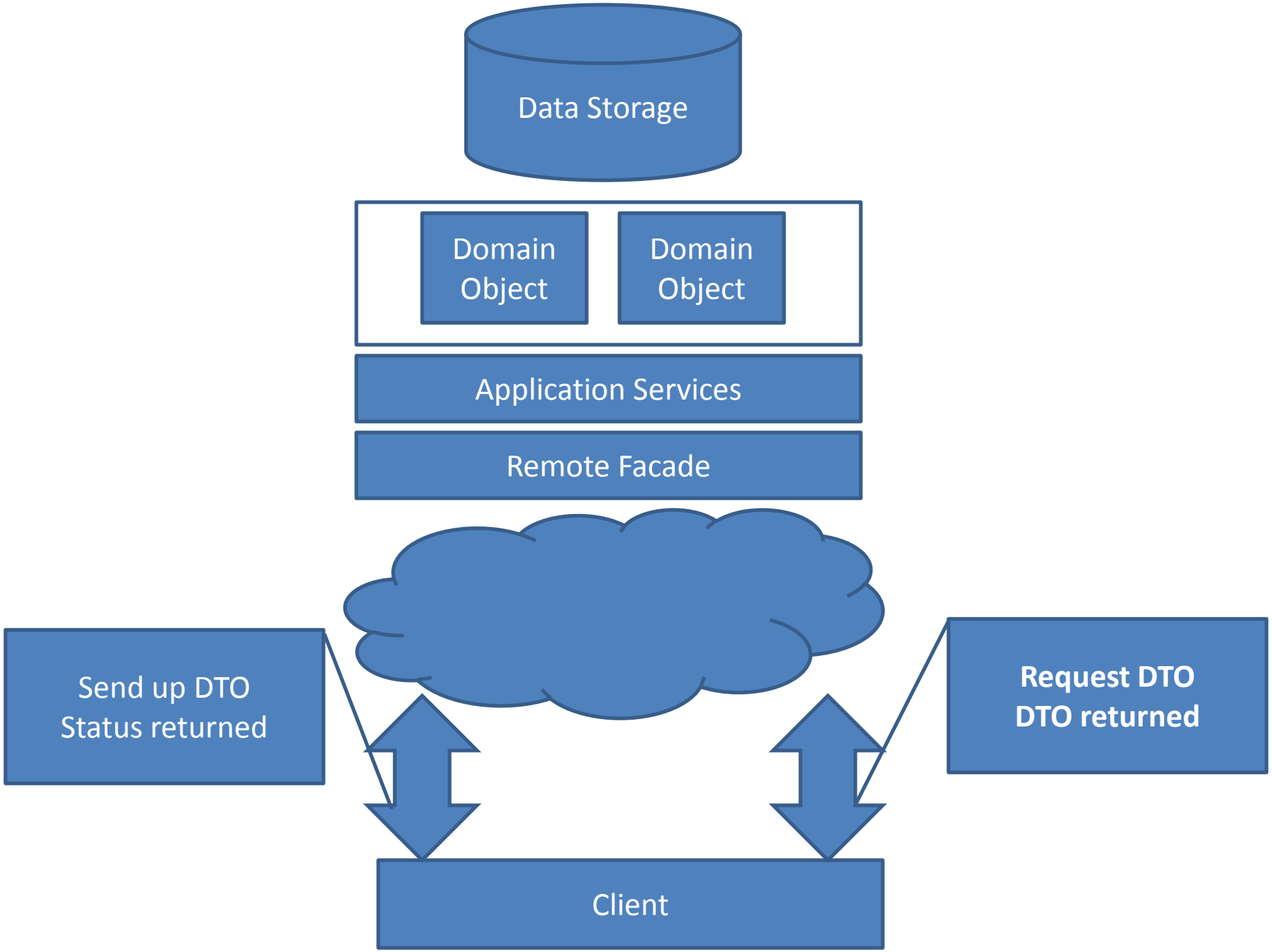

AddItemCommand
ChangeAddressCommand
Purchase Order

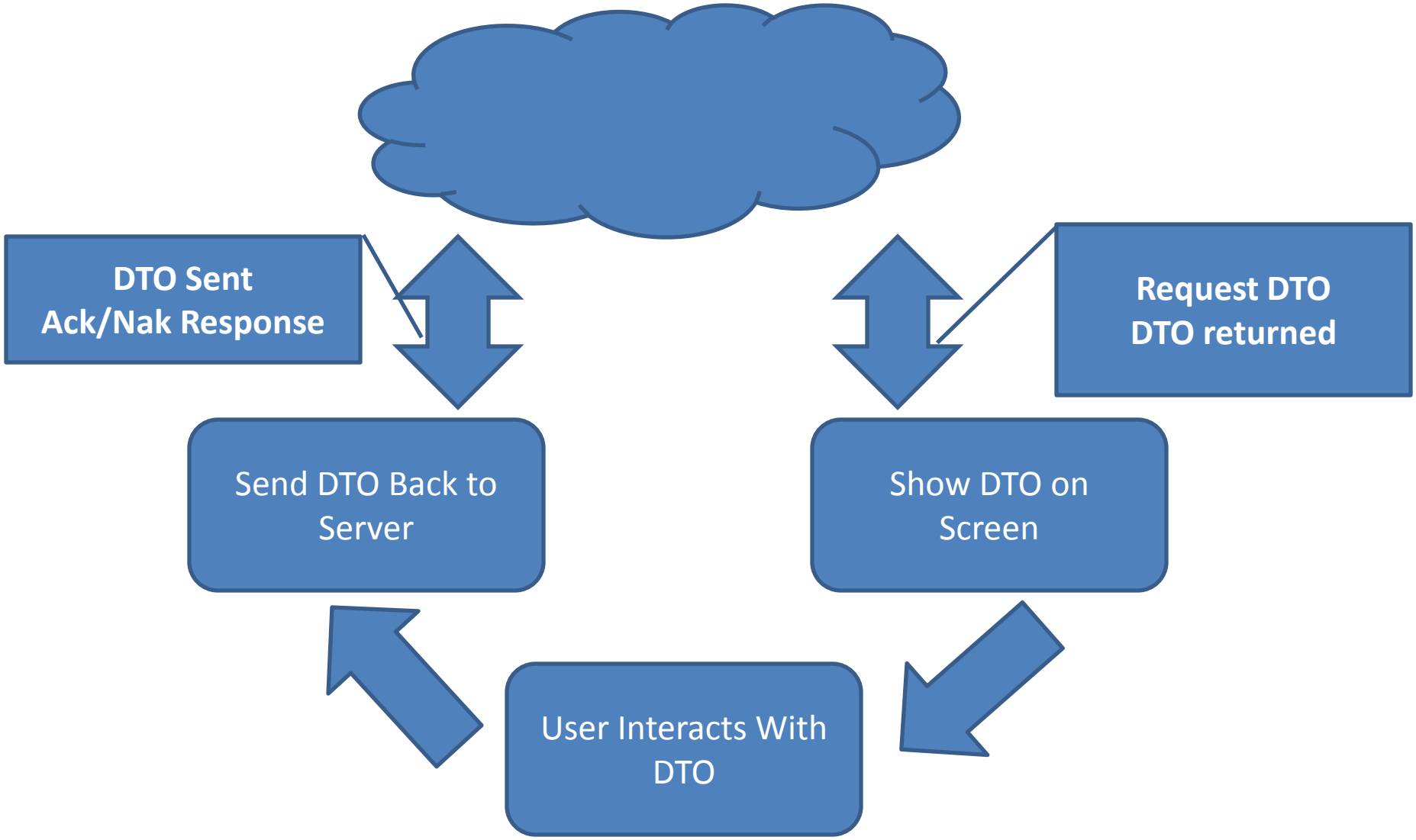

Cart Created
3 Items Added
1Item Removed
Shipping Information Added

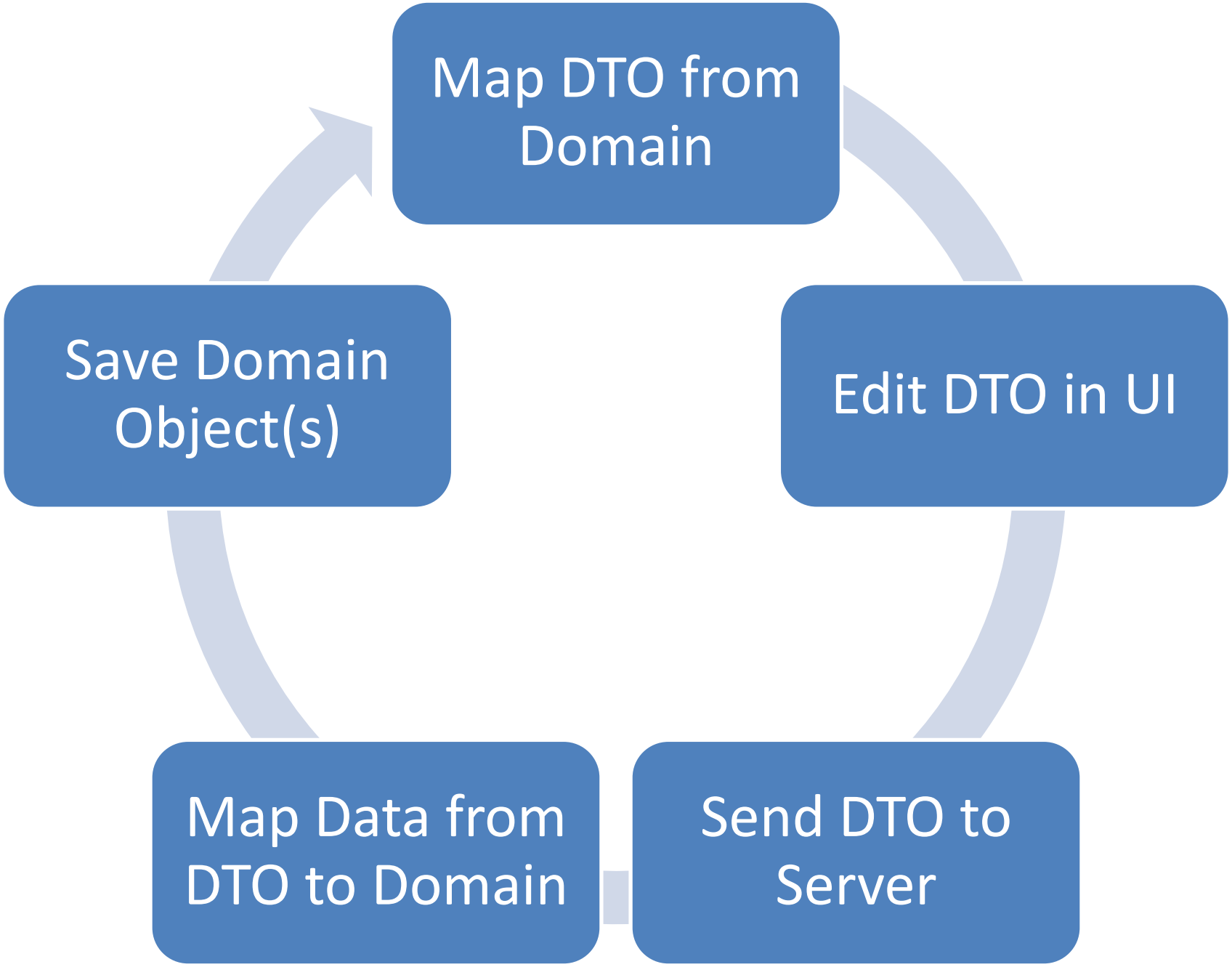

Replay
Skip

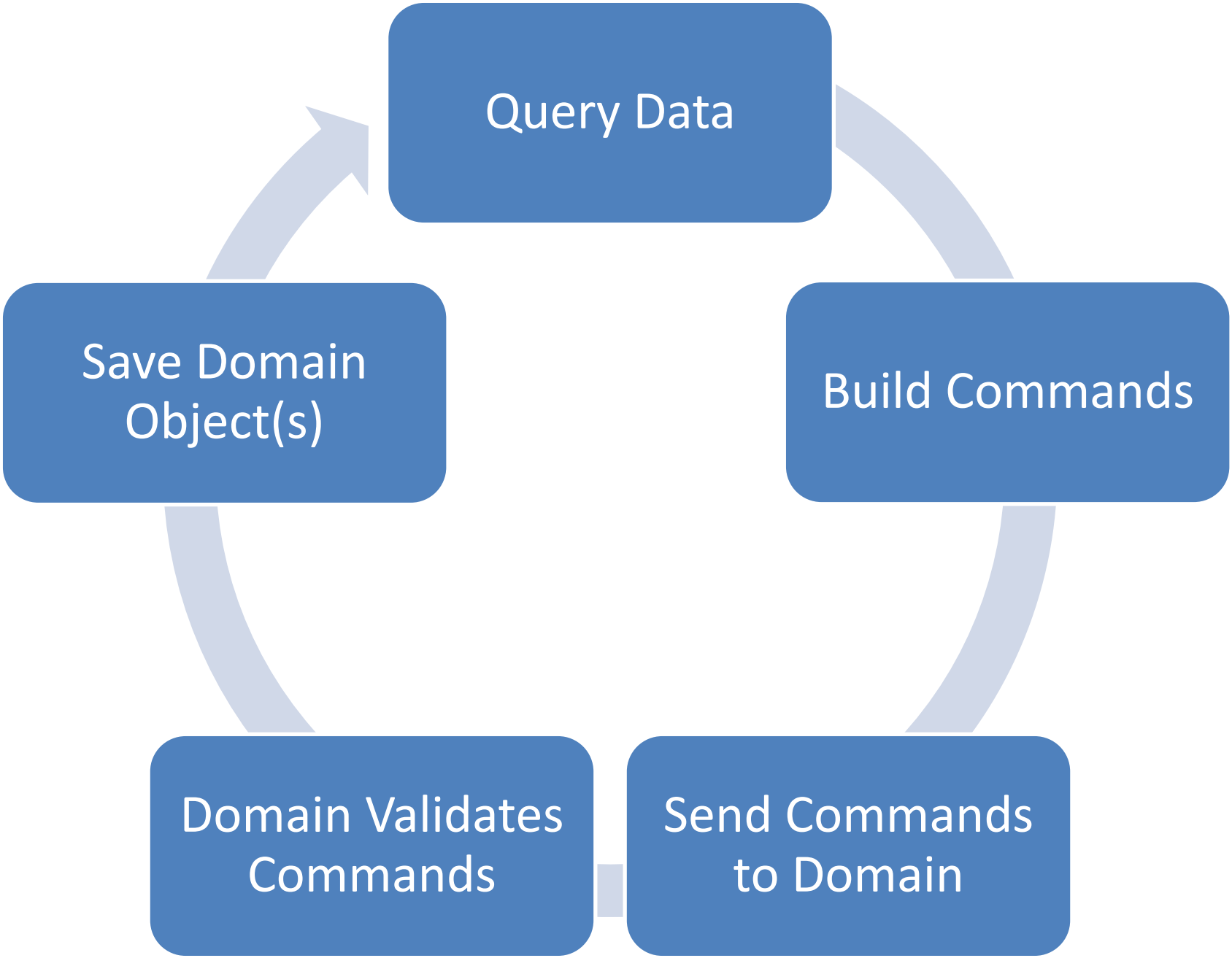

7
6
5
4
3
2
1

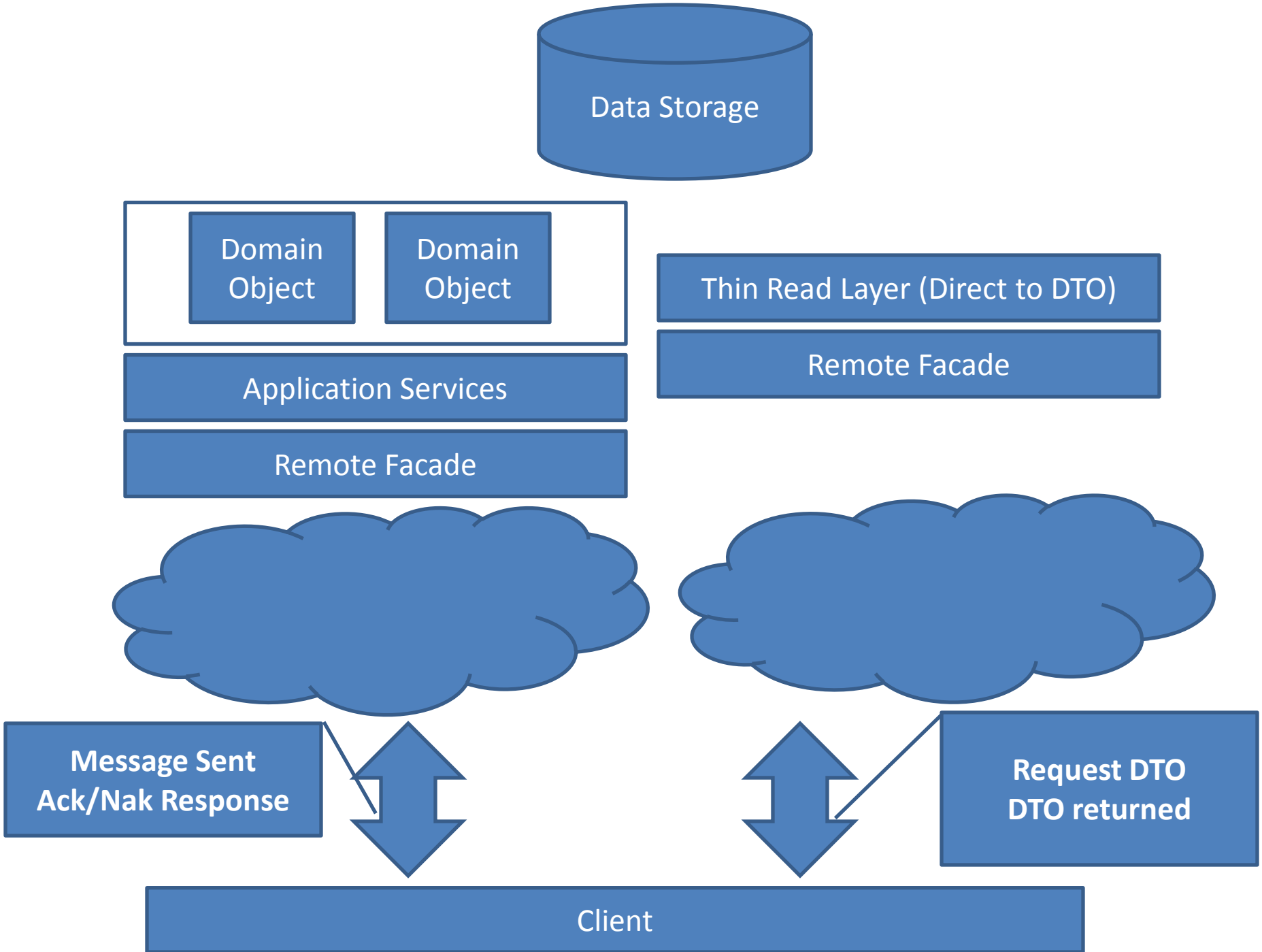

6
5
snap
4
3
2
1

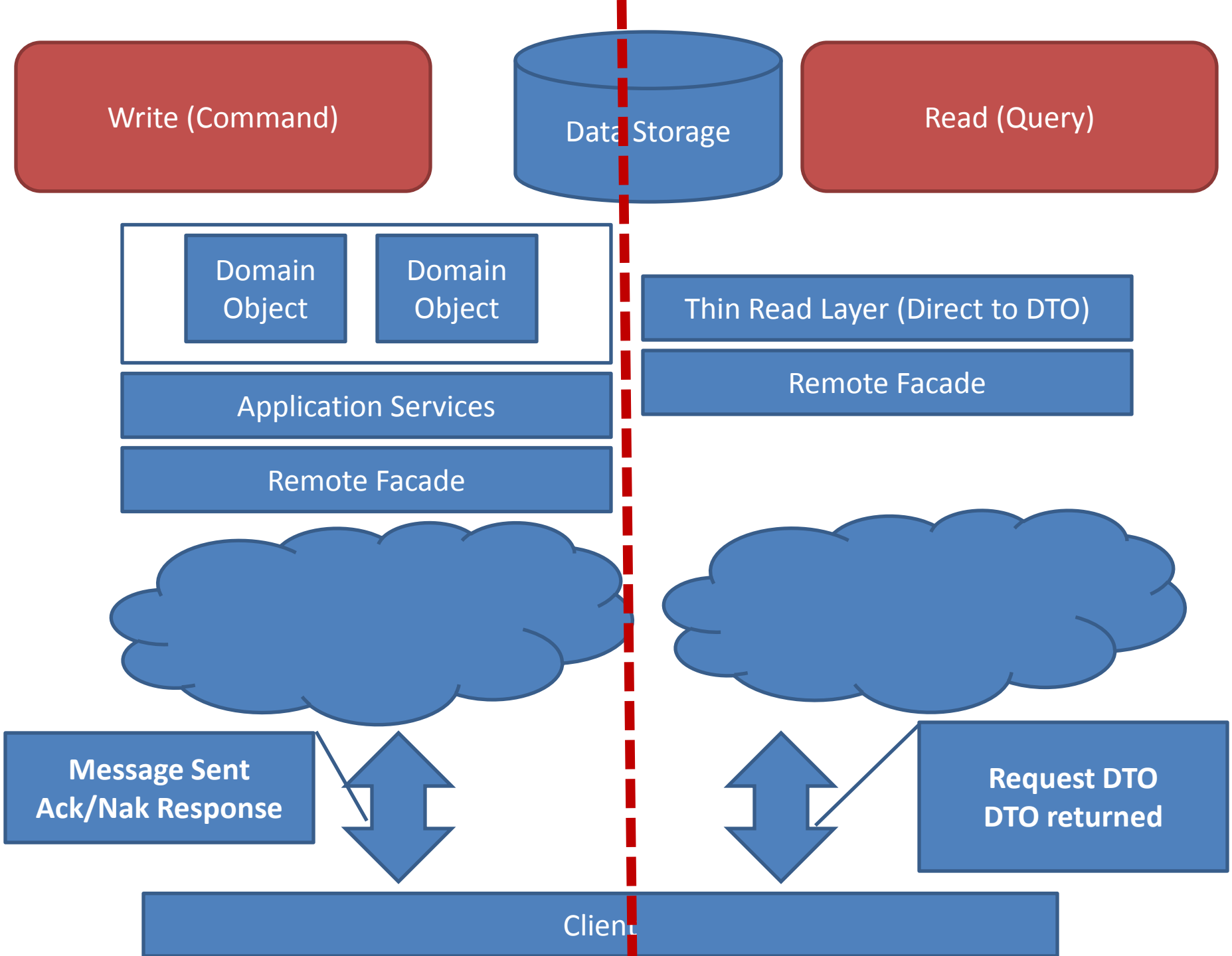

Data Storage
Domain Object
Domain Object
Application Services
Remote Facade
Send up DTO
Status returned
Request DTO
DTO returned
Client


DTO Sent
Ack/Nak Response
Request DTO
DTO returned
Send DTO Back to Server
Show DTO on Screen
User Interacts With DTO

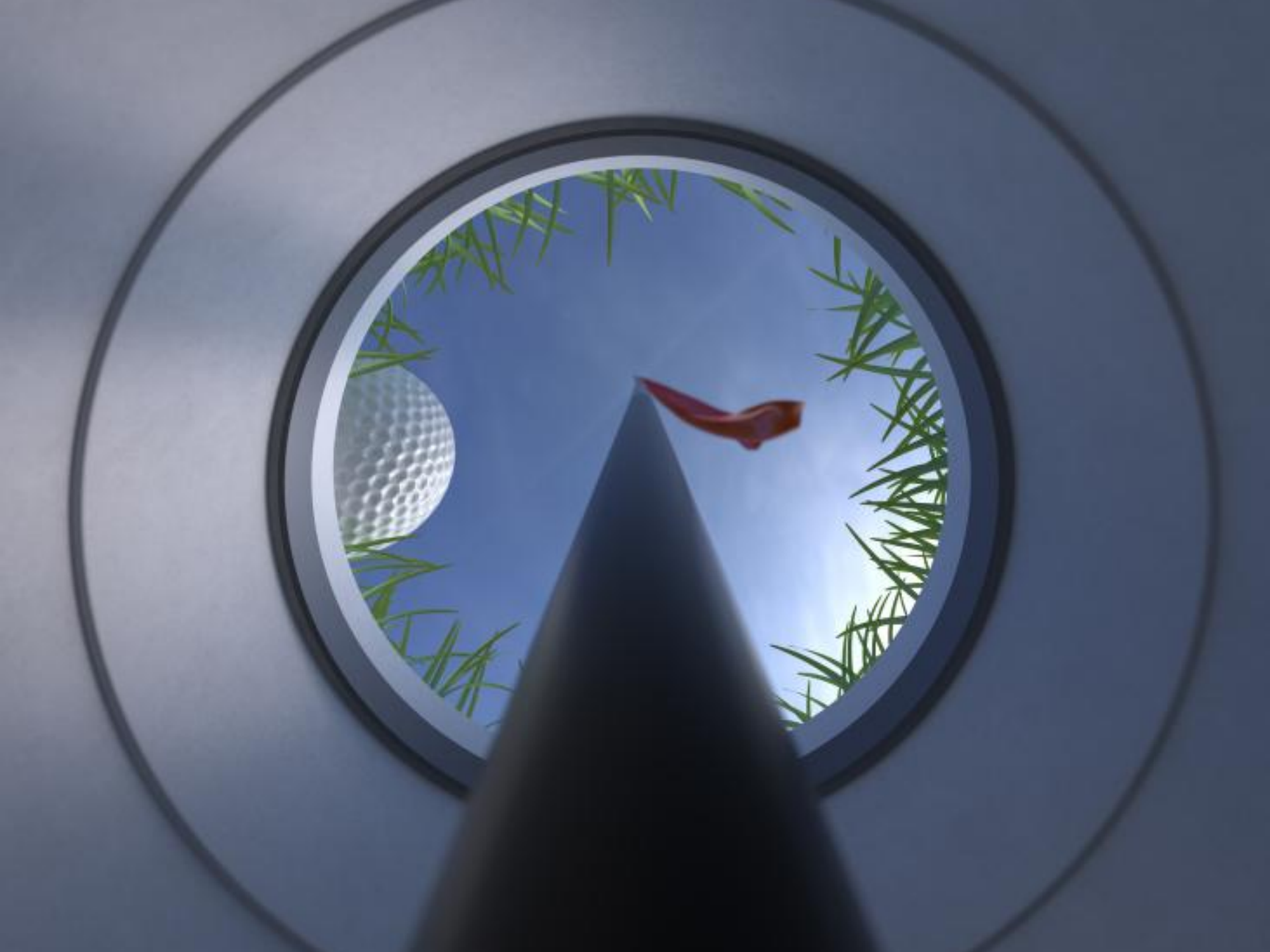

Map DTO from Domain
Edit DTO in UI
Send DTO to Server
Map Data from DTO to Domain
Save Domain Object(s)


Query Data
Build Commands
Send Commands to Domain
Domain Validates Commands
Save Domain Object(s)


Data Storage
Domain Object
Domain Object
Application Services
Remote Facade
Thin Read Layer (Direct to DTO)
Remote Facade
Message Sent
Ack/Nak Response
Request DTO
DTO returned
Client

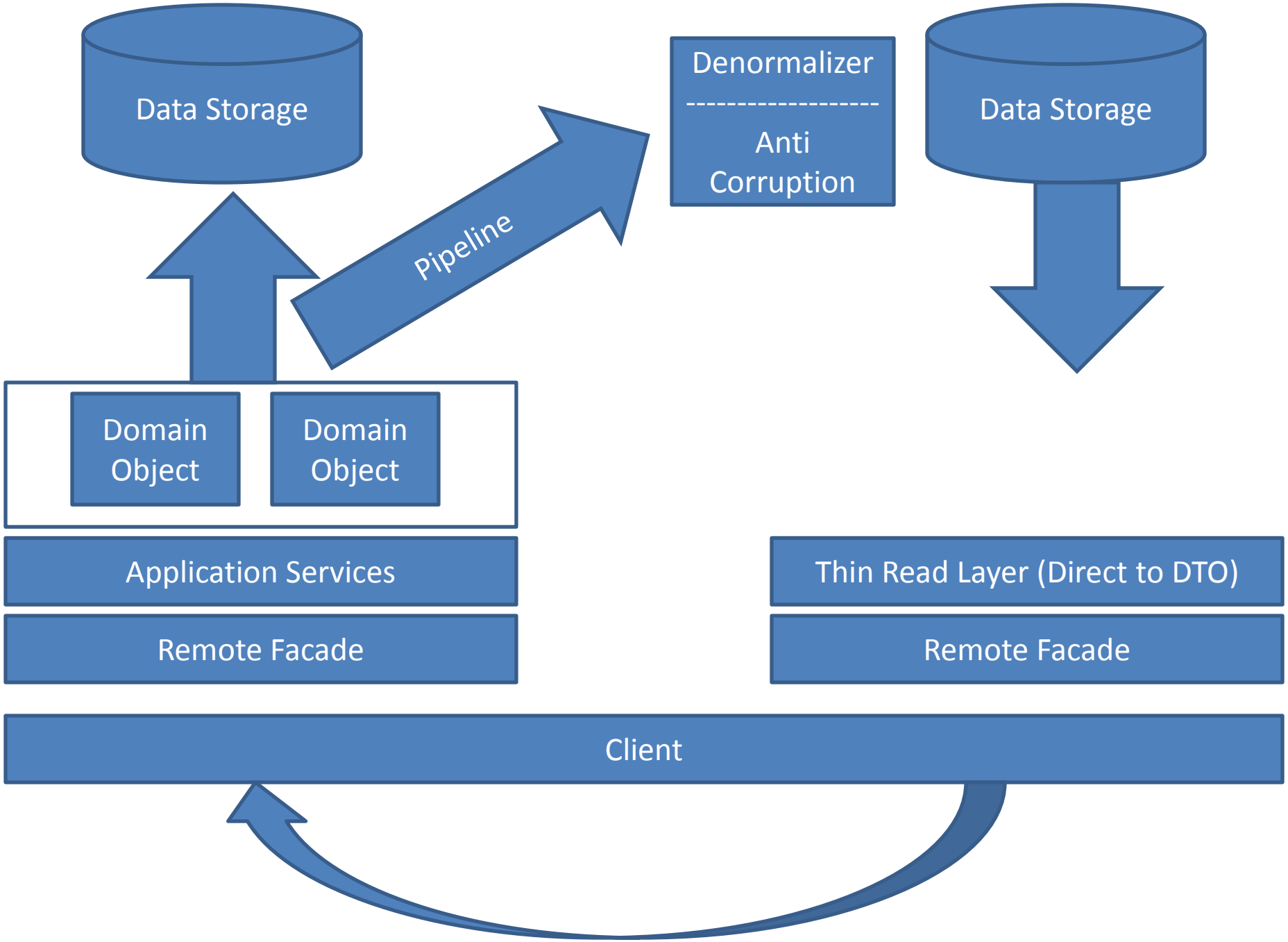

Write (Command)
Data Storage
Read (Query)
Domain Object
Domain Object
Application Services
Remote Facade
Thin Read Layer (Direct to DTO)
Remote Facade
Message Sent
Ack/Nak Response
Request DTO
DTO returned
Client

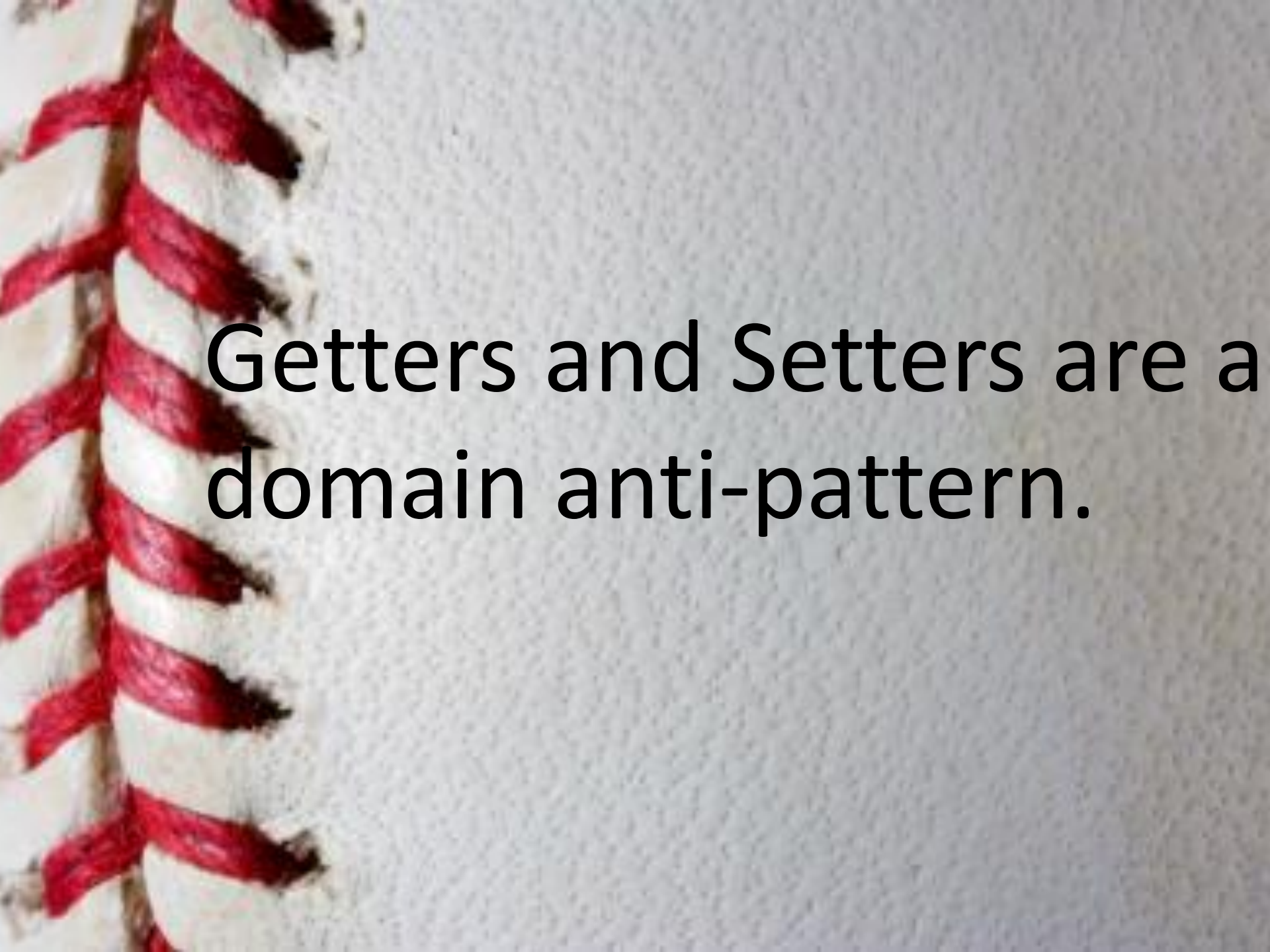
A single model cannot be appropriate for reporting, searching, and transactional behaviors…



The model that a client needs the data in a distributed system is screen based and different than the domain model.

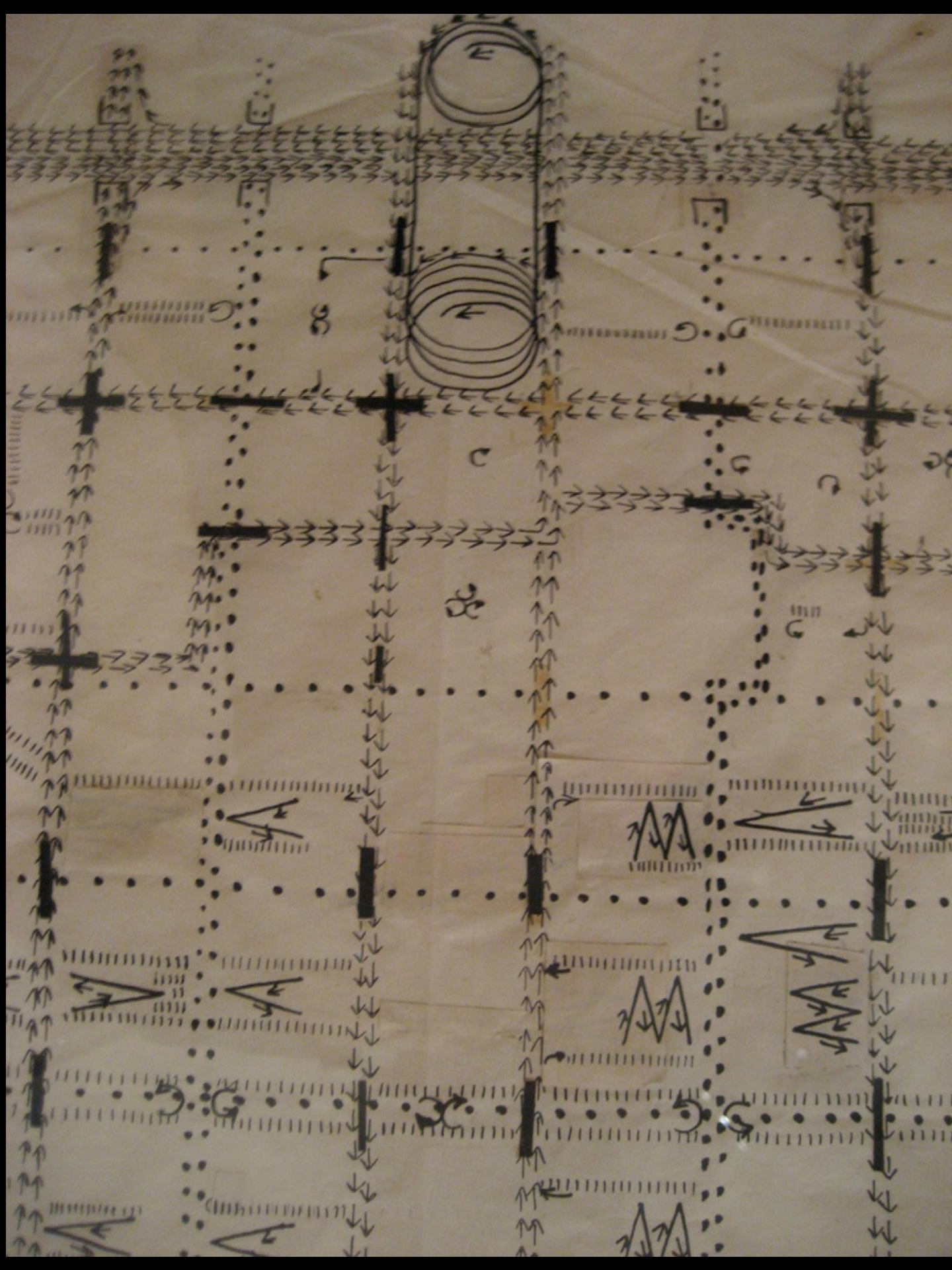
Most queries can operate with relaxed consistency…


Data Storage
Denormalizer
Anti Corruption
Data Storage
Pipeline
Domain Object
Domain Object
Application Services
Remote Facade
Thin Read Layer (Direct to DTO)
Remote Facade
Client


Getters and Setters are a
domain anti-pattern.

```
public interface Content extends IdBasedDomainObject, Editable {
    String getByline();
    DateTime getLastLiveTime();
    DateTime getPublicationDate();
    DateTime getWebPublicationDateTime();
    Publication getPublication();
    Integer getPageNumber();
    boolean isSensitive();
    String getMarkedSensitiveBy();
    DateTime getMarkedSensitiveOn();
    List<Tag> getTags();
    List<Keyword> getKeywords();
    NewspaperBook getBook();
    NewspaperBookSection getBookSection();
    Series getSeries();
    Contributor getContributor();
    List<Contributor> getContributors();
    List<Tone> getTones();
    Section getDerivedSection();
    Page getPage();
    List<Content> getNestedContent();
    boolean shouldBeDeletedWhenPageIsDeleted();
    boolean isTrailblockFromDisplayStoryPackage();
    boolean isPluckCommentable();
    boolean isSynchronisedWithPluck();
    void removeSeries();
    void addTag(Tag tag);
    void removeTag(Tag tag);
    void replaceTag(Tag originalTag, Tag replacementTag);
    boolean isTagSetValid();
    List<Series> getSeriesList();
    String getTypeName();
    String getTrailNameDisplay();
    Trailblock<TrailblockElement> getTrailblock();
    boolean hasSensitiveKeyword();
    void setFootballMatchReference(ExternalReference externalReference);
    ExternalReference getFootballMatchReference();
    void setCricketMatchReference(ExternalReference externalReference);
    ExternalReference getCricketMatchReference();
    ExternalReference getFilmReference();
    void setFilmReference(ExternalReference filmsExternalReference);
    void setFilm(Film film);
    Film getFilm();
    DateTime getScheduledExpiryDate();
    boolean isExpired();
    DateTime getClosingDateForCommenting();
    DateTime getClosingDateForCommentRecommending();
    boolean isCommentingClosed();
    boolean isCommentRecommendingClosed();
    StarRating getStarRating();
    boolean isInMicrosite();
    List<? extends Factbox> getFactboxes();
    boolean hasTone(ToneName name);
    boolean isEditorial();
    boolean isBlockAds();
```

```
public interface Content extends IdBasedDomainObject, Editable {
    String getByline();
    DateTime getLastLiveTime();
    DateTime getPublicationDate();
    DateTime getWebPublicationDateTime();
    Publication getPublication();
    Integer getPageNumber();
    boolean isSensitive();
    String getMarkedSensitiveBy();
    DateTime getMarkedSensitiveOn();
    List<Tag> getTags();
    List<Keyword> getKeywords();
    NewspaperBook getBook();
    NewspaperBookSection getBookSection();
    Series getSeries();
    Contributor getContributor();
    List<Contributor> getContributors();
    List<Tone> getTones();
    Section getDerivedSection();
    Page getPage();
    List<Content> getNestedContent();
    boolean shouldBeDeletedWhenPageIsDeleted();
    boolean isTrailblockFromDisplayStoryPackage();
    boolean isPluckCommentable();
    boolean isSynchronisedWithPluck();
    void removeSeries();
    void addTag(Tag tag);
    void removeTag(Tag tag);
    void replaceTag(Tag originalTag, Tag replacementTag);
    boolean isTagSetValid();
    List<Series> getSeriesList();
    String getTypeName();
    String getTrailNameDisplay();
    Trailblock<TrailblockElement> getTrailblock();
    boolean hasSensitiveKeyword();
    void setFootballMatchReference(ExternalReference externalReference);
    ExternalReference getFootballMatchReference();
    void setCricketMatchReference(ExternalReference externalReference);
    ExternalReference getCricketMatchReference();
    ExternalReference getFilmReference();
    void setFilmReference(ExternalReference filmsExternalReference);
    void setFilm(Film film);
    Film getFilm();
    DateTime getScheduledExpiryDate();
    boolean isExpired();
    DateTime getClosingDateForCommenting();
    DateTime getClosingDateForCommentRecommending();
    boolean isCommentingClosed();
    boolean isCommentRecommendingClosed();
    StarRating getStarRating();
    boolean isInMicrosite();
    List<? extends Factbox> getFactboxes();
    boolean hasTone(ToneName name);
    boolean isEditorial();
    boolean isBlockAds();
```



Most Bounded Contexts can interact with relaxed consistency.


60
5
10
15
20
25
30
35
40
45
50
55

Using relaxed consistency allows us to increase our scalability and availability!

“Man acts as though he were the shaper and master of language, while in fact it is language that is the master of man.”

State transitions are an important part of our problem space and should be modeled within our domain.

Getters and Setters are a domain smell.

Most Bounded Contexts can interact with relaxed consistency.

A single model cannot be appropriate for reporting, searching, and transactional behaviors…

Questions