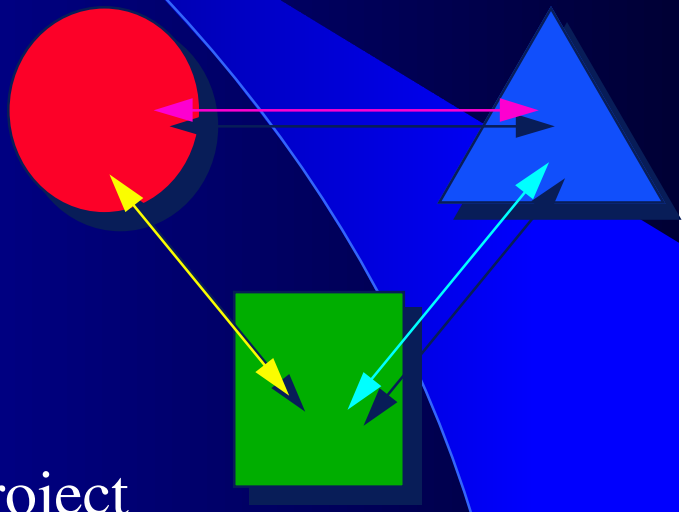
Common Object Services

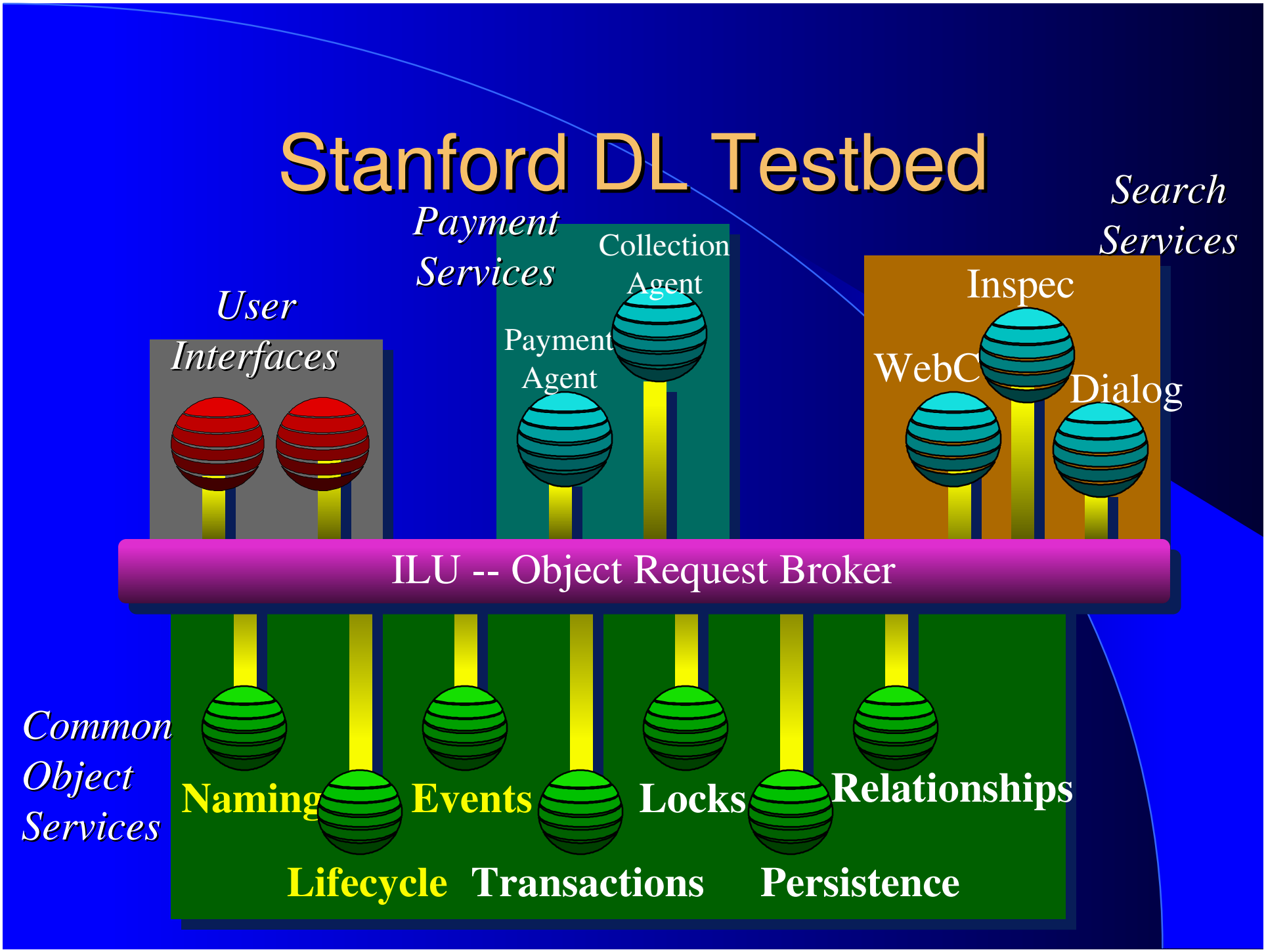


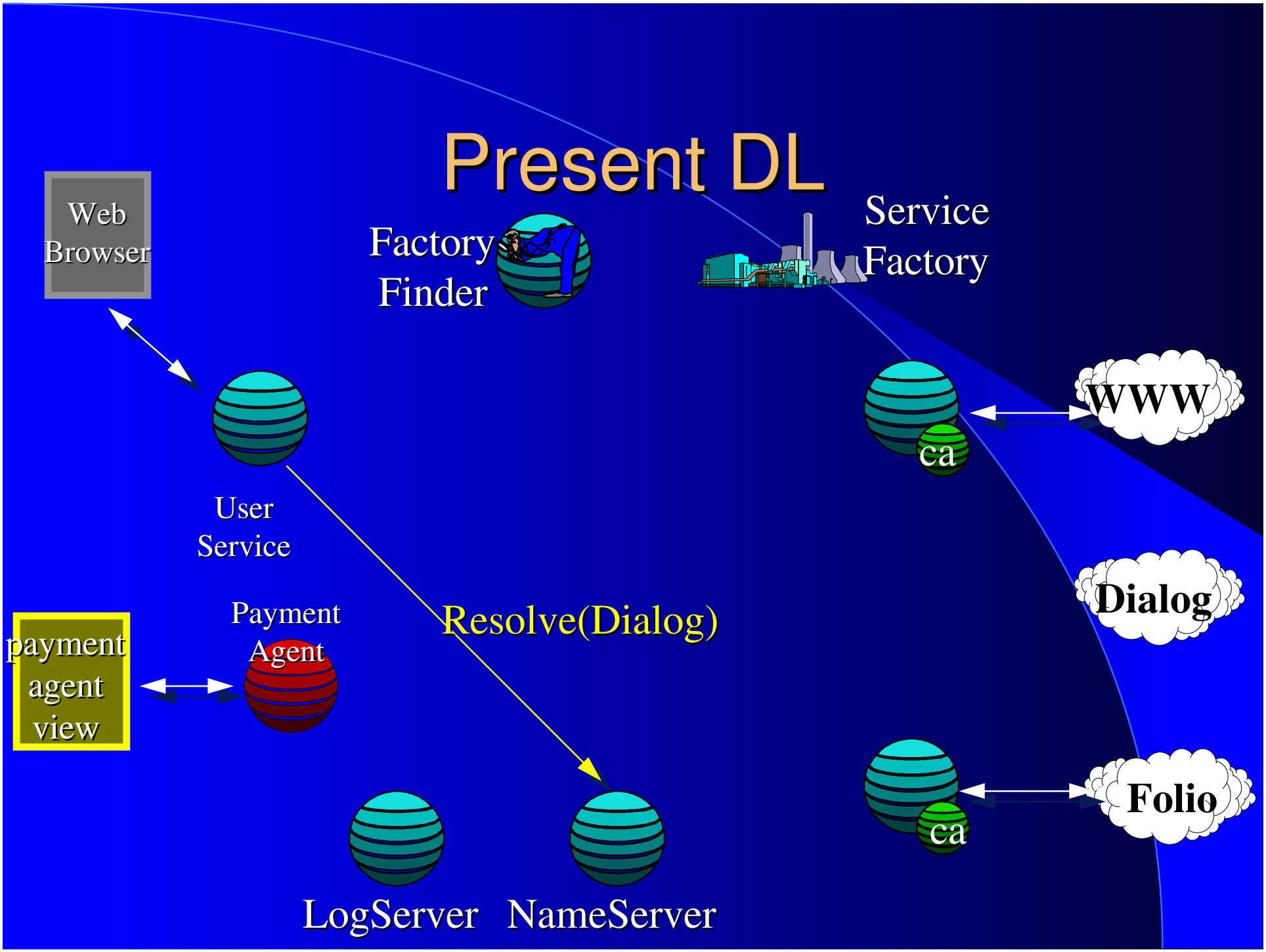
Scott W. Hassan

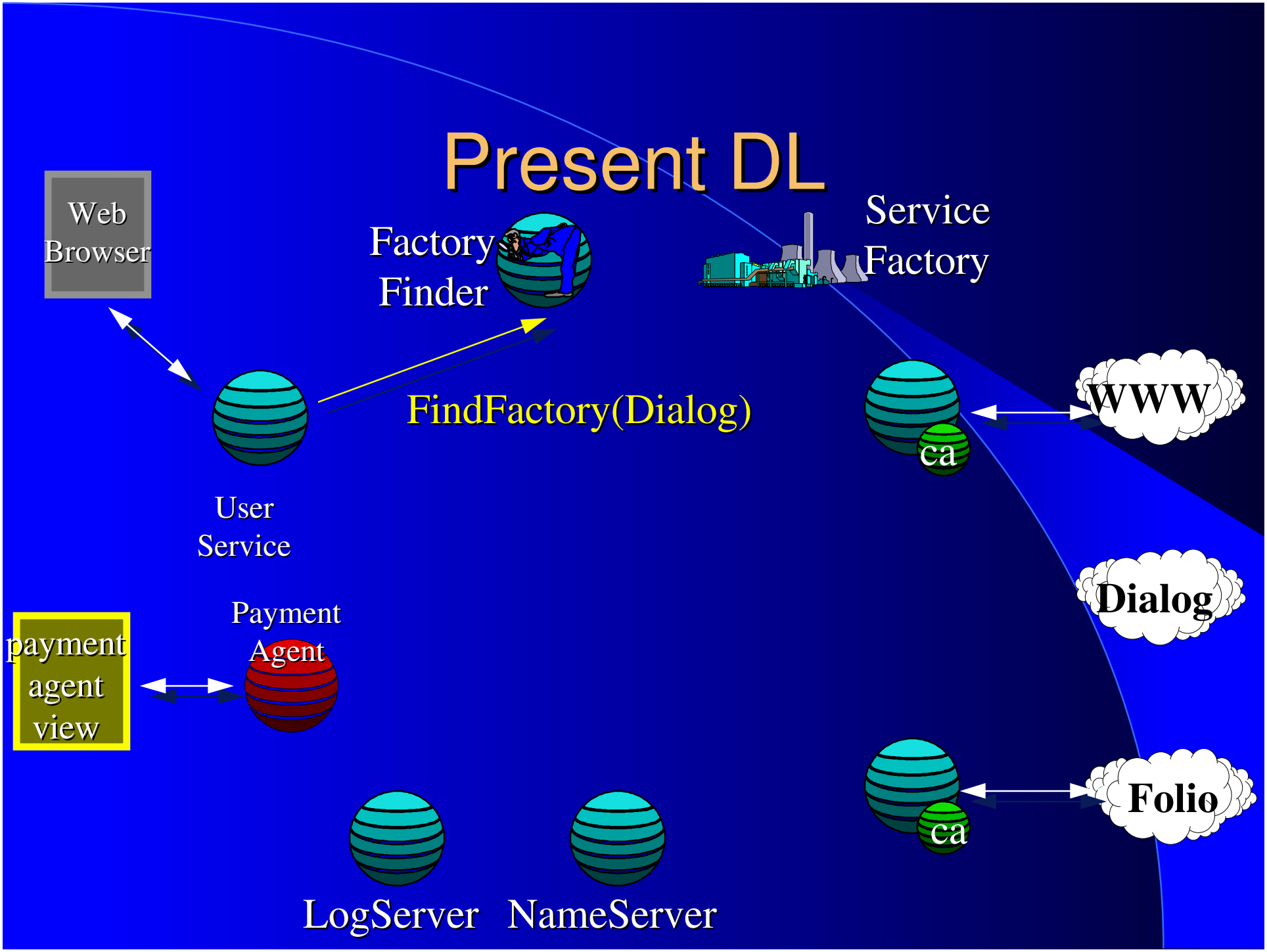
Stanford Digital Library Project

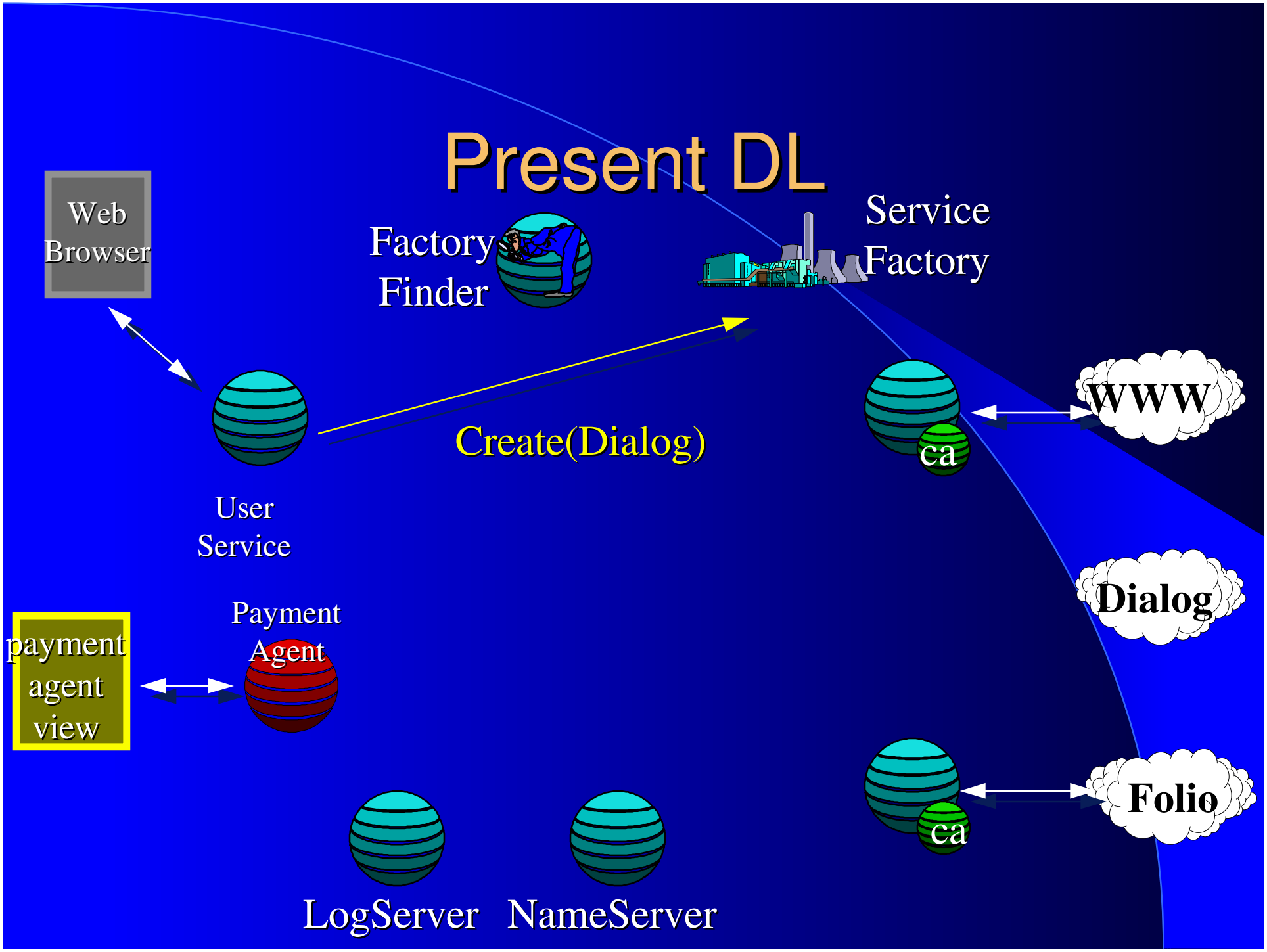
Stanford University

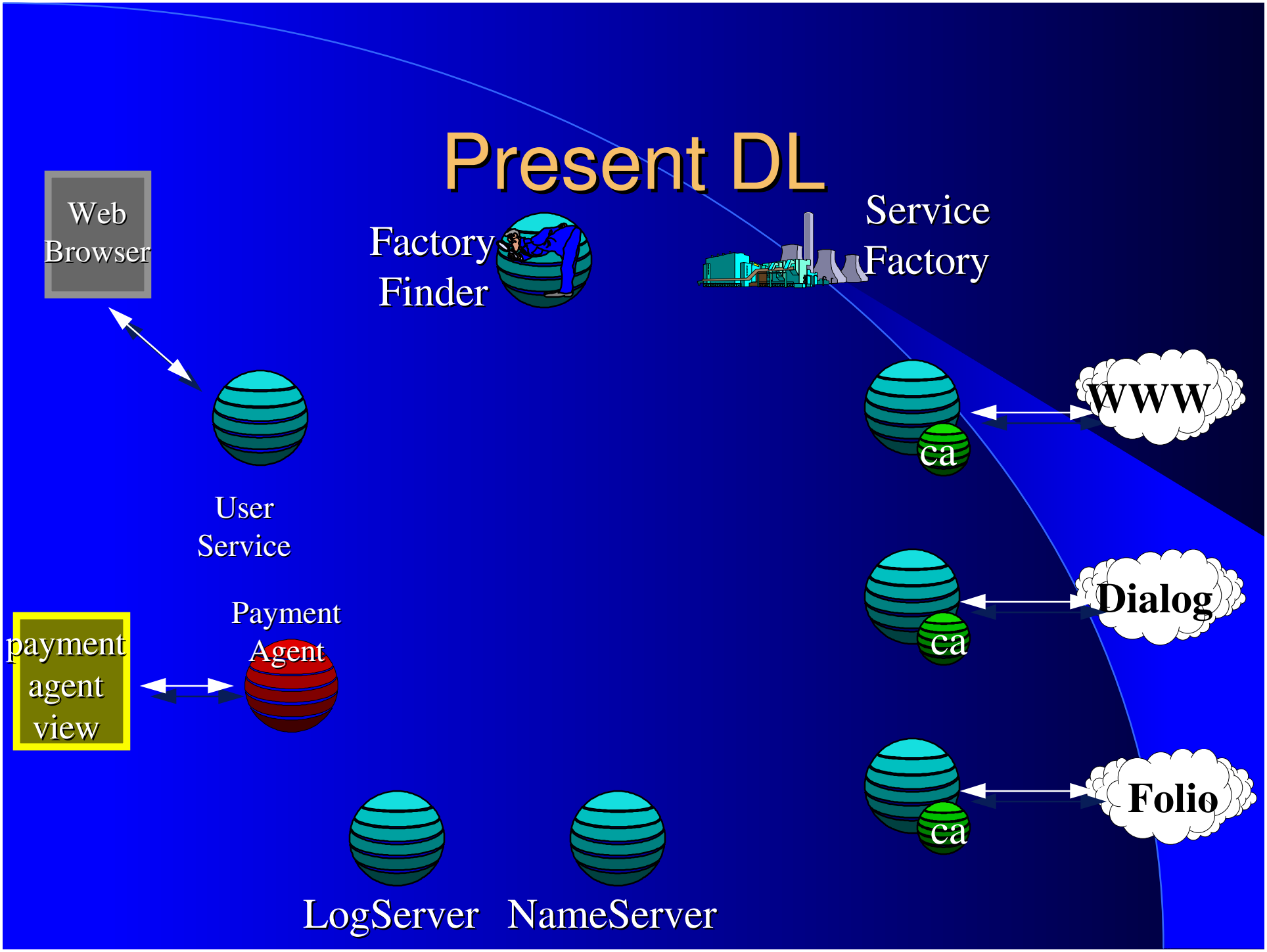
Stanford DL Testbed

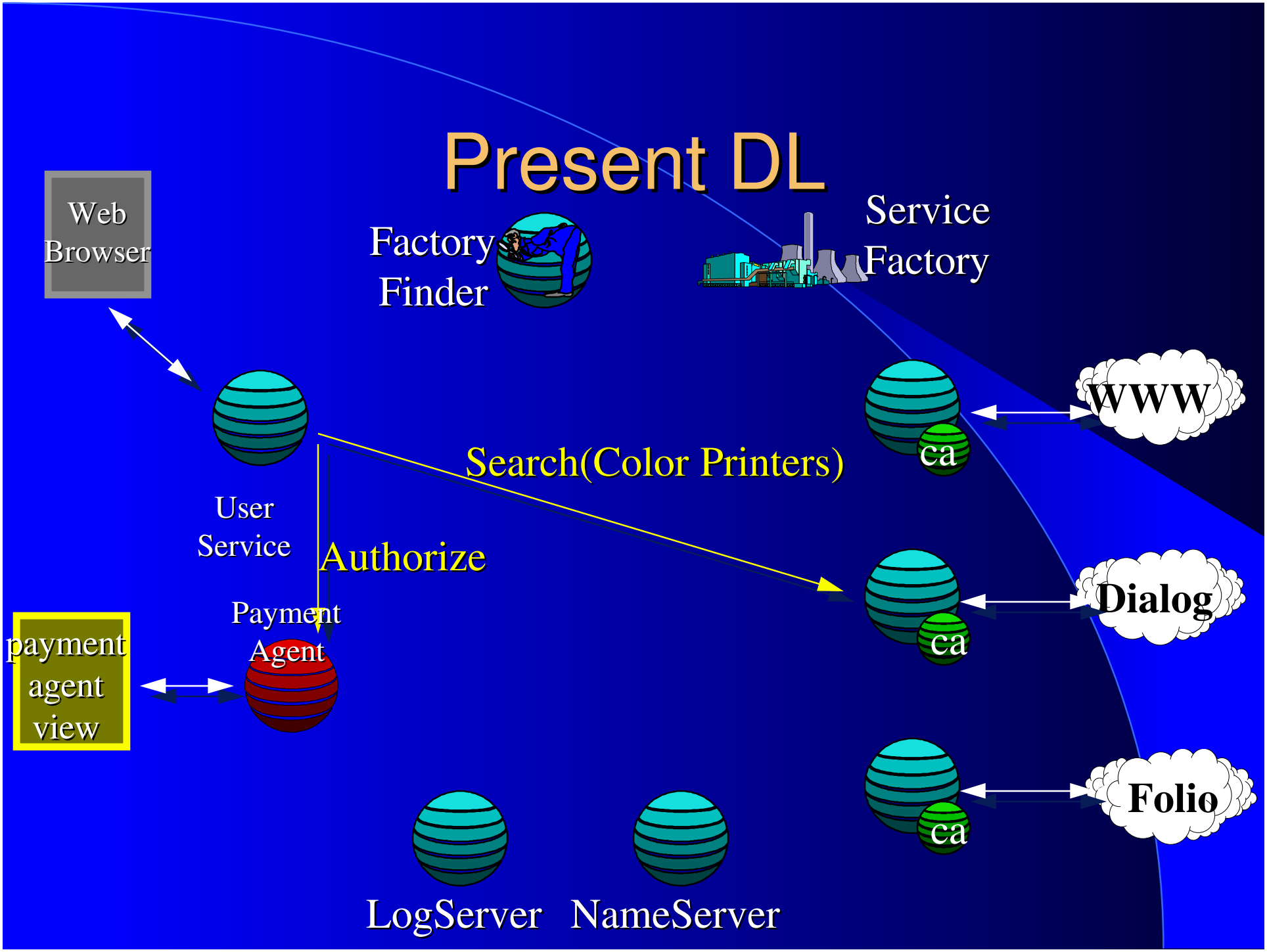


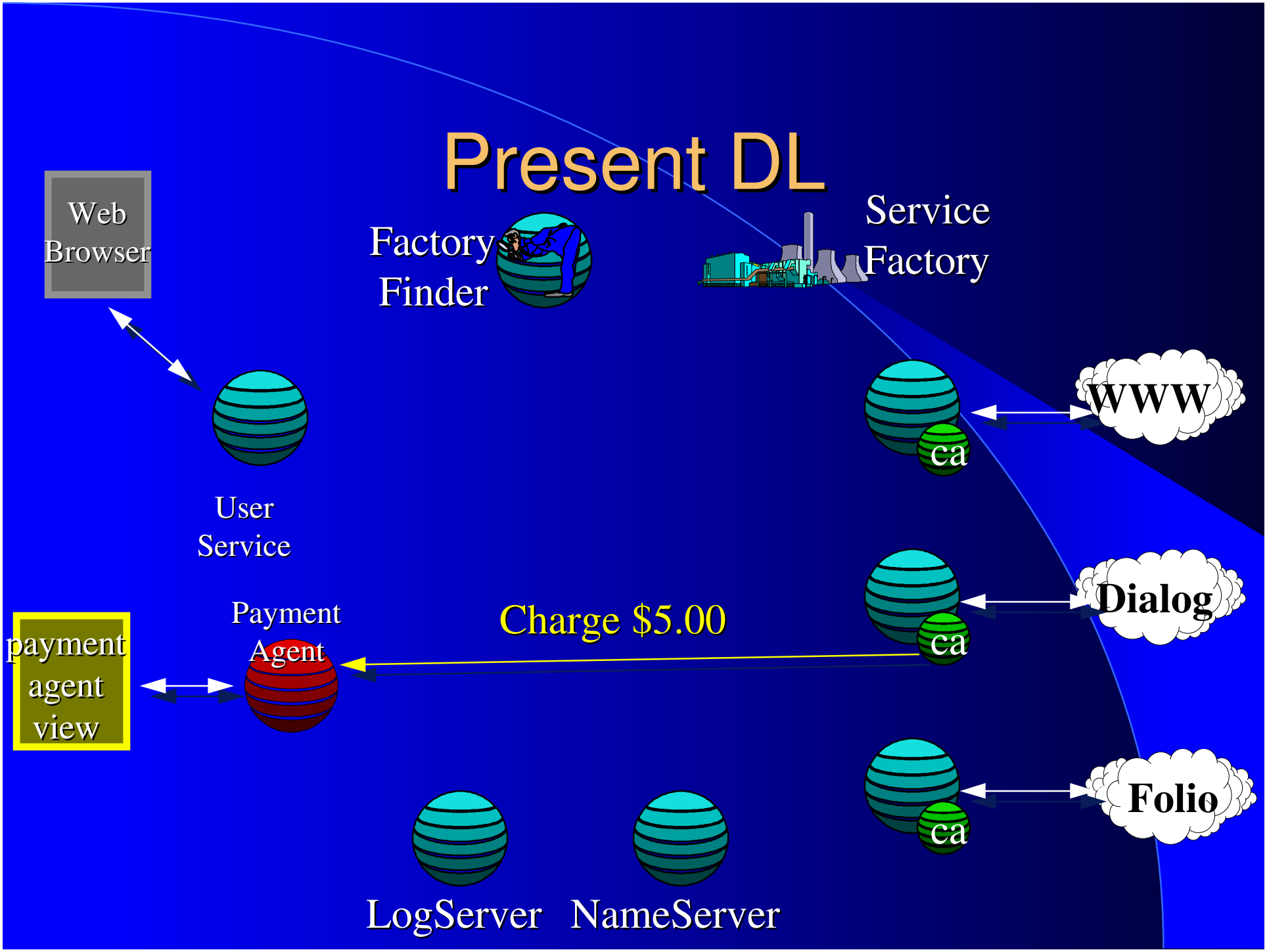

Present DL
Web Browser
Factory Finder
Service Factory
User Service
ca
WWW
Dialog
payment agent view
Payment Agent
Resolve(Dialog)
ca
Folio
LogServer
NameServer

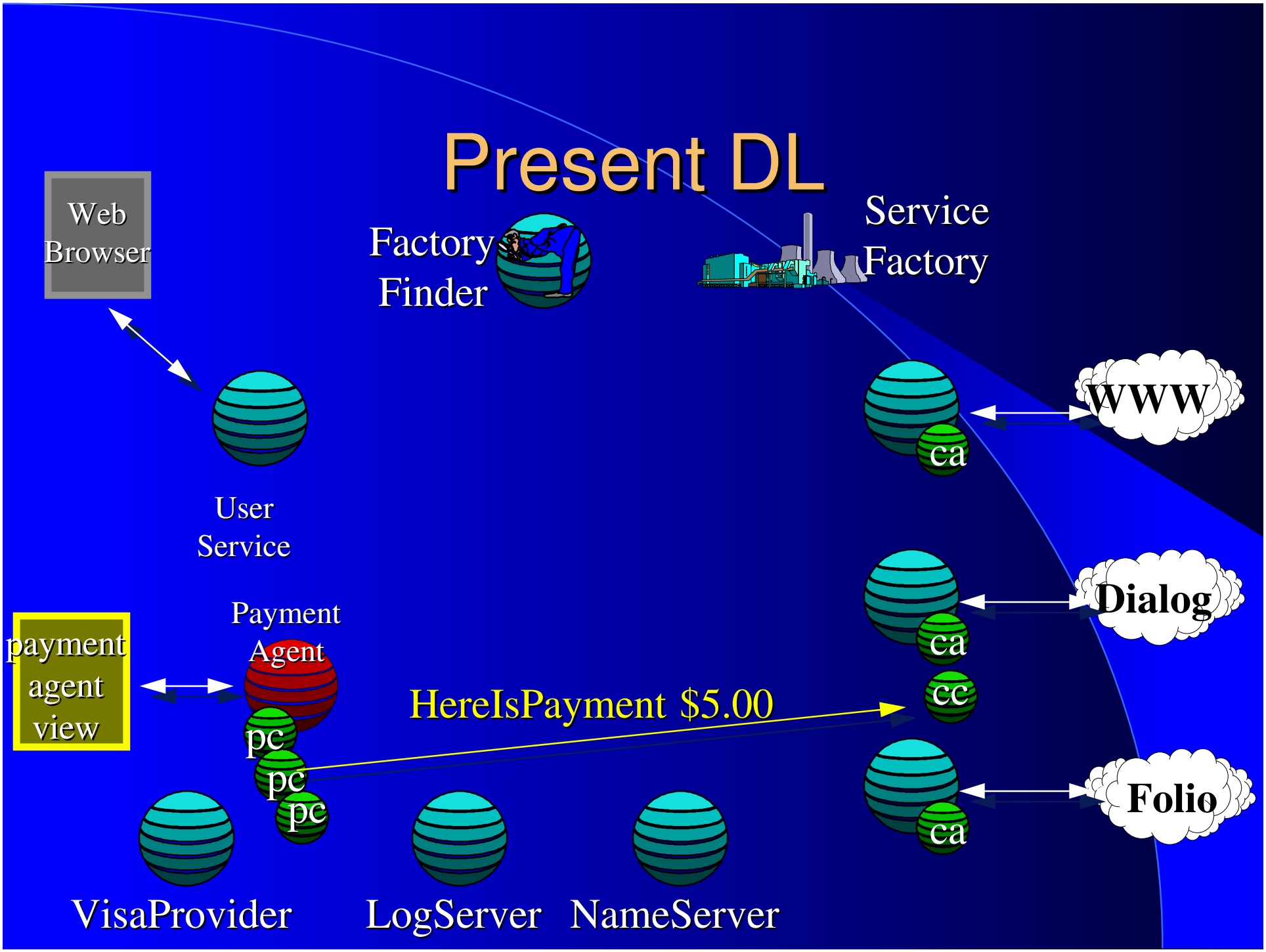

Present DL
Web Browser
Factory Finder
Service Factory
FindFactory(Dialog)
User Service
ca
WWW
Dialog
payment agent view
Payment Agent
ca
Folio
LogServer
NameServer

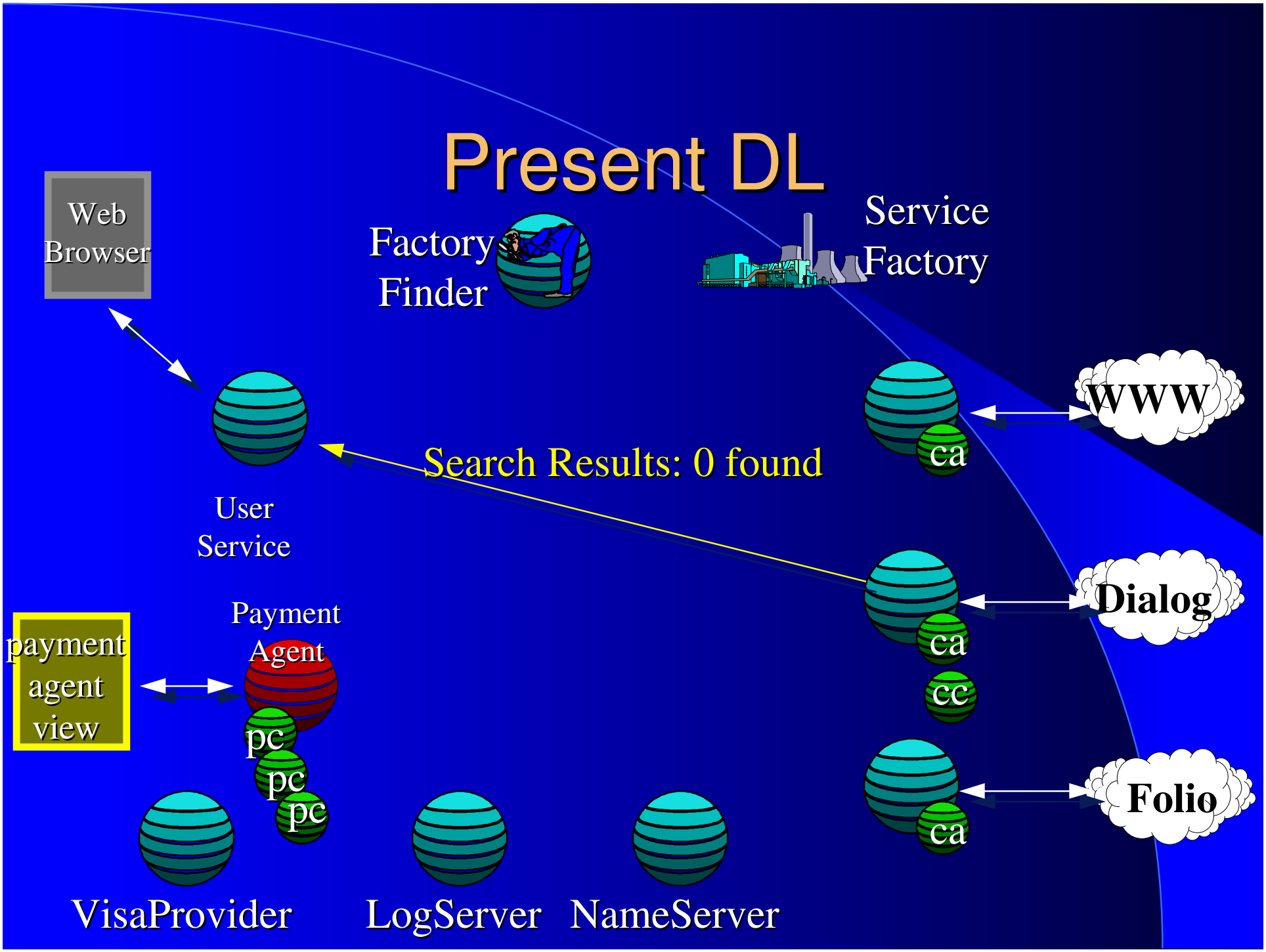

Present DL
Web Browser
Factory Finder
Service Factory
Create(Dialog)
User Service
ca
WWW
Dialog
payment agent view
Payment Agent
ca
Folio
LogServer
NameServer

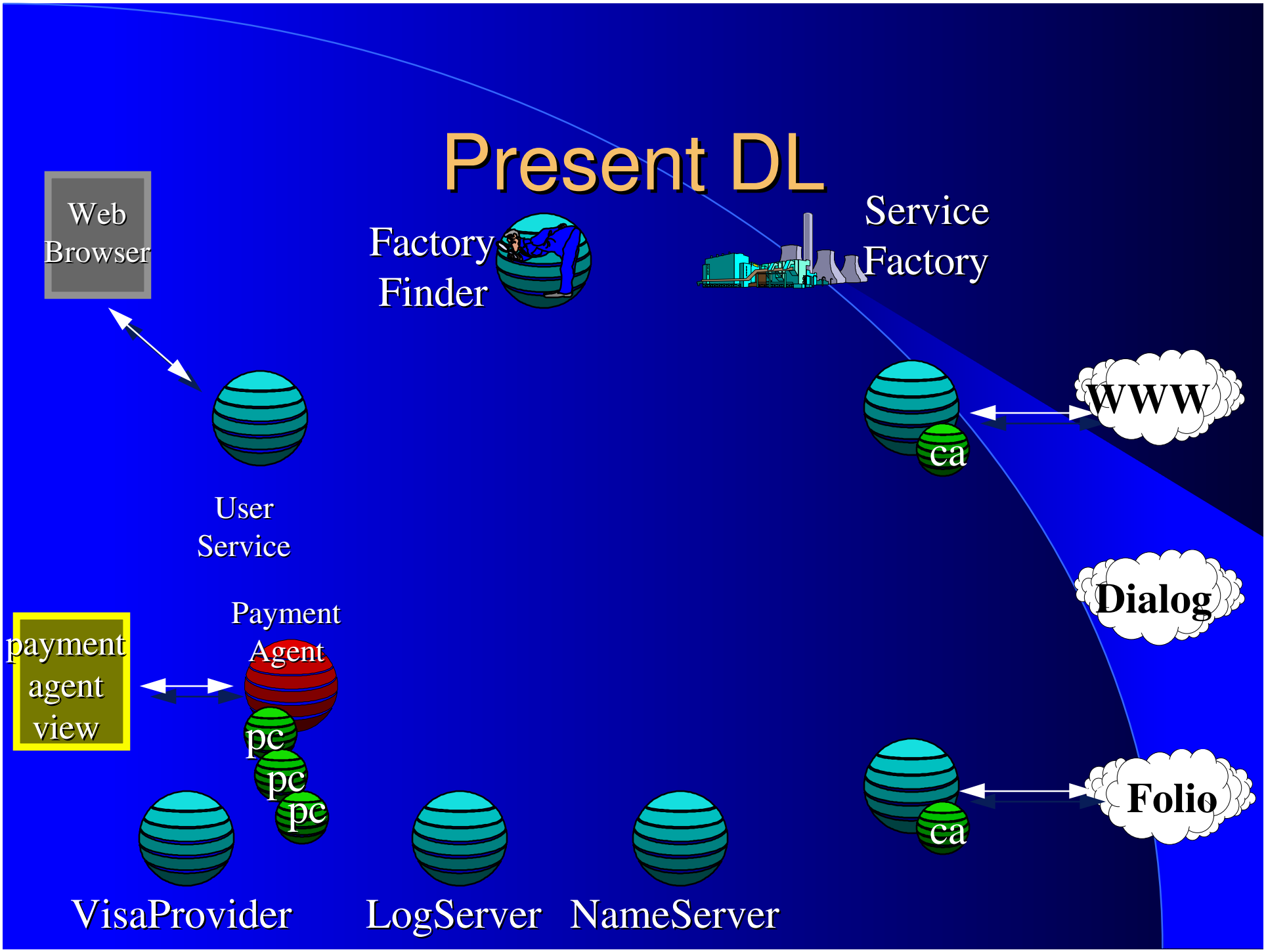

Present DL
Web Browser
Factory Finder
Service Factory
User Service
ca
WWW
Payment Agent
payment agent view
ca
Dialog
ca
Folio
LogServer
NameServer


Present DL
Web Browser
Factory Finder
Service Factory
WWW
ca
Search(Color Printers)
User Service
Authorize
Dialog
ca
Payment Agent
payment agent view
Folio
ca
LogServer
NameServer

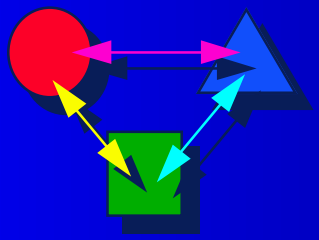

Present DL
Web Browser
Factory Finder
Service Factory
User Service
ca
WWW
Payment Agent
Charge $5.00
payment agent view
ca
Dialog
ca
Folio
LogServer
NameServer


Present DL
Web Browser
Factory Finder
Service Factory
User Service
ca
WWW
Payment Agent
payment agent view
HereIsPayment $5.00
pc
pc
pc
ca
cc
Dialog
ca
Folio
VisaProvider
LogServer
NameServer

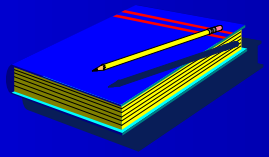

Present DL
Web Browser
Factory Finder
Service Factory
WWW
ca
Search Results: 0 found
User Service
Dialog
payment agent view
Payment Agent
ca
cc
pc
pc
pc
Folio
ca
VisaProvider
LogServer
NameServer


Present DL
Web Browser
Factory Finder
Service Factory
User Service
ca
WWW
Dialog
payment agent view
Payment Agent
pc
pc
pc
ca
Folio
VisaProvider
LogServer
NameServer

Agenda



Common Object Services



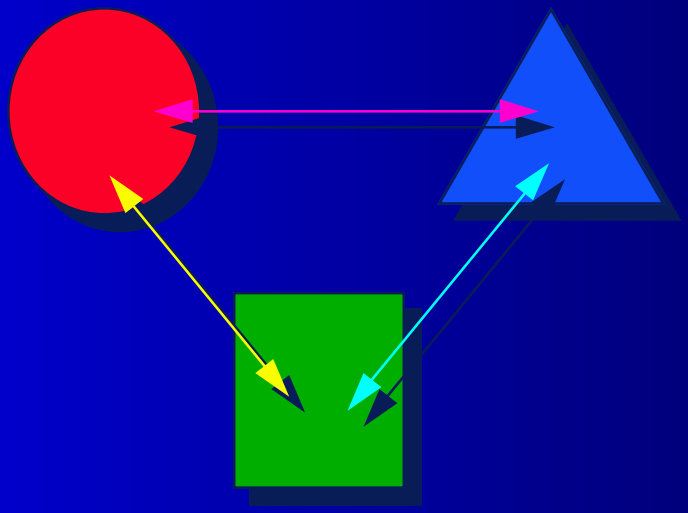
Name Service

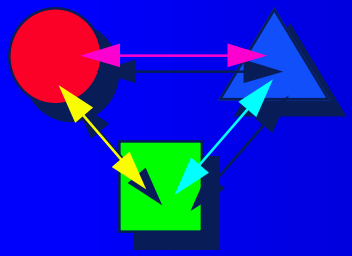
Event Service

Life Cycle Service

Common Object Services

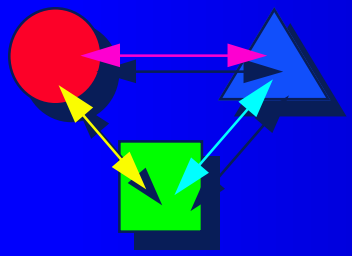
- “Fundamental services that are critical for creating and maintaining objects within a distributed computing environment.” - OMG





Service

- An object or group of objects that collaborate to provide function.
  - “building blocks”
  - could be distributed across networks

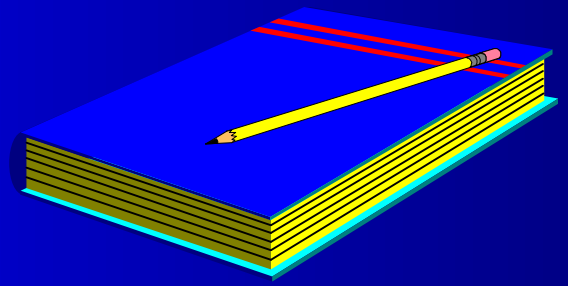


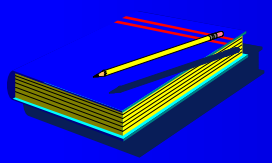
Some COS

- Archive, Backup/Restore, Change Management, Concurrency Control, Data Interchange, **Event**, Externalization, Implementation Repository, Installation and Activation, Interface Repository, Licensing, **Lifecycle**, **Naming**, Operational Control, Persistance, Properties, Query, Relationships, Replication, Security, Startup Services, Threads, Time, Trading, Transactions.

Name Service

- **Bind** - associate a name to an object
- **Resolve** - lookup an object given a name

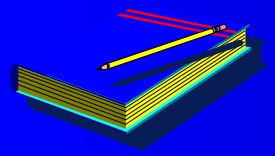


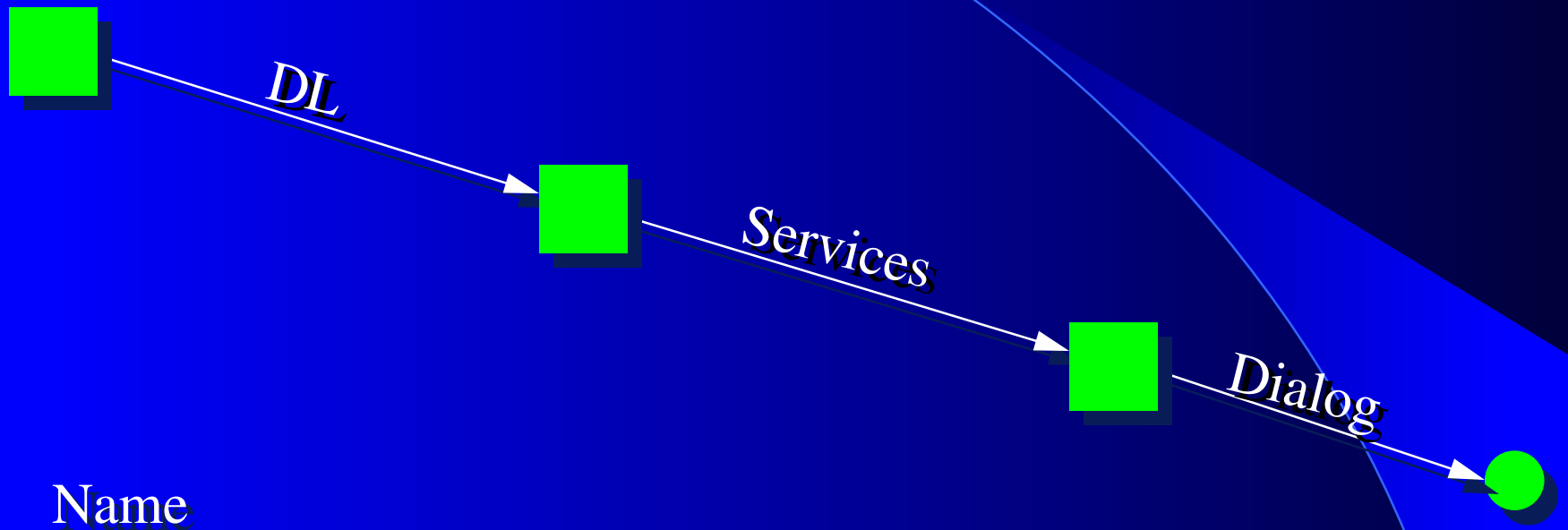


Name Service Architecture

- distributed
- heterogeneous
- composed of names and naming contexts
  - directed graph with cycles
  - names are relative to a naming context
  - no absolute names

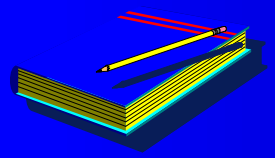
Name Example





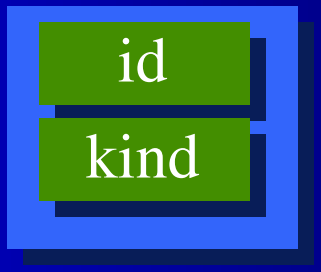
Name

DL	Services	Dialog
Naming Context	Naming Context	Object

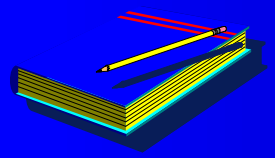


Name

- “no semantics or interpretation of the names”
- TName - Ordered sequence of components
- TNameComponent

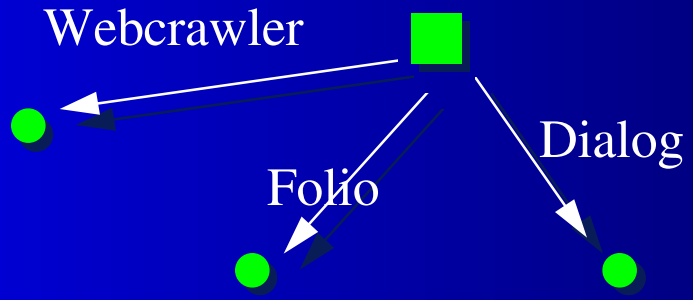


- “/DL/Services/Dialog”

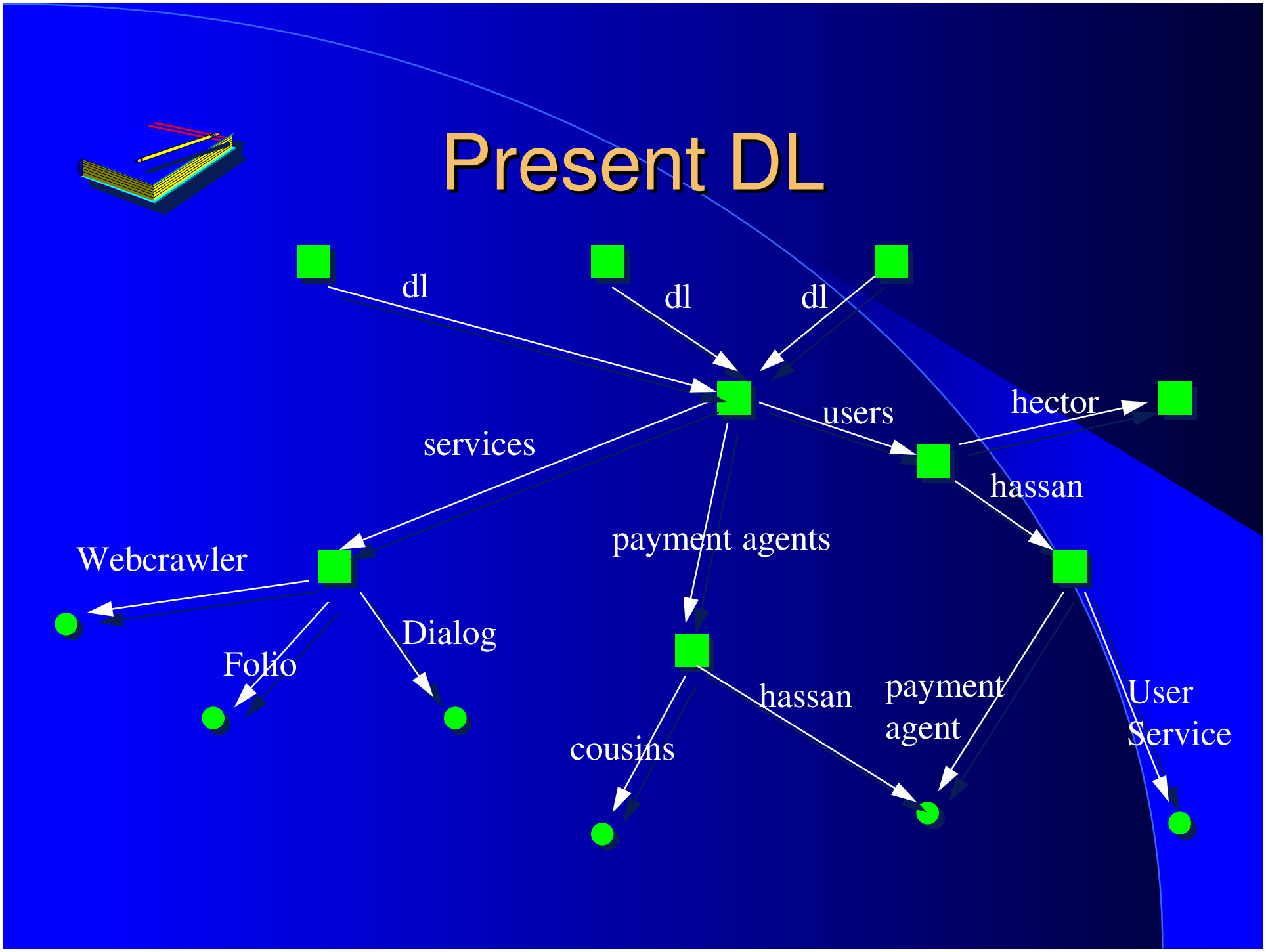


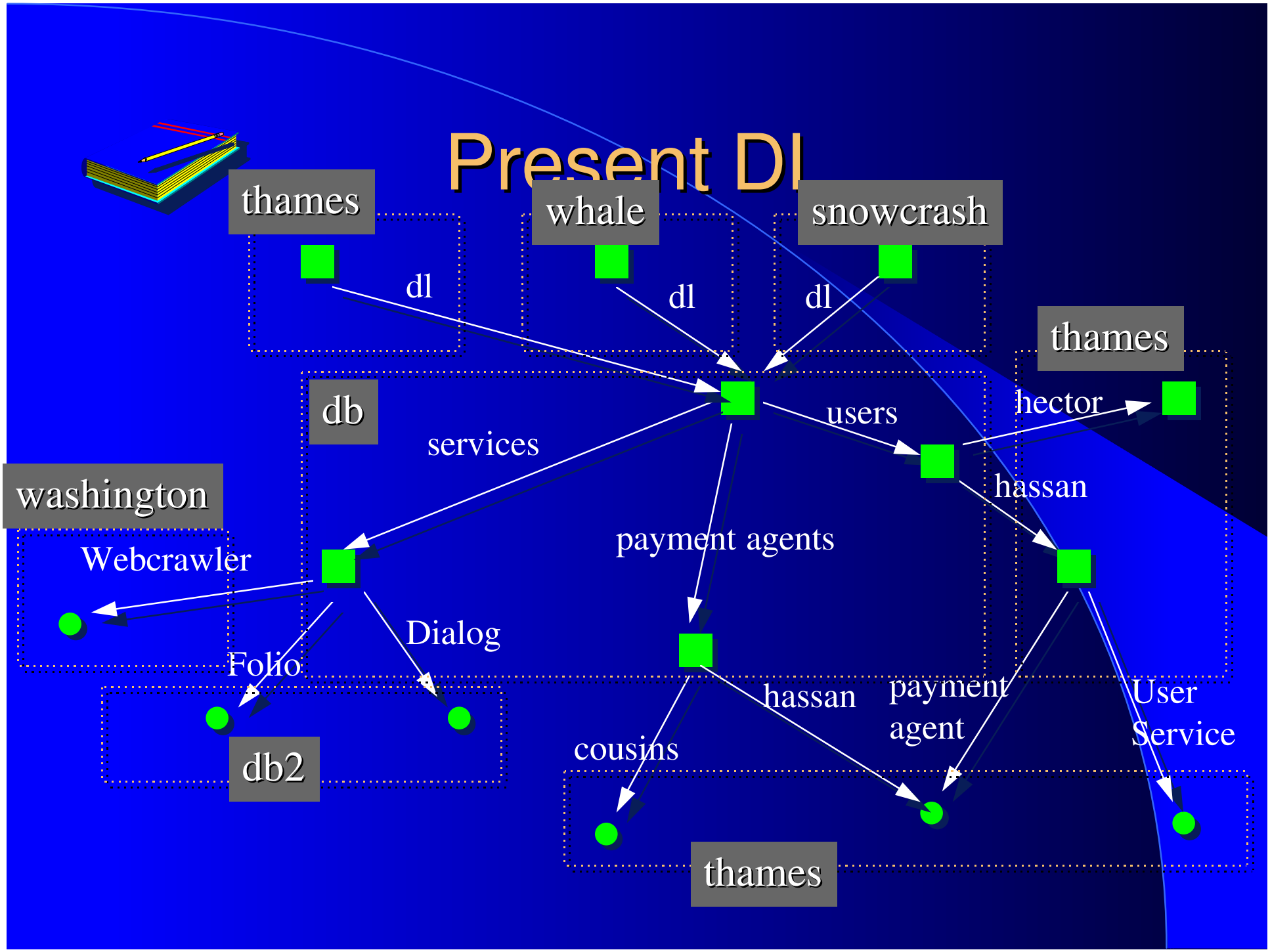
Naming Context

- a set of name bindings
  - TName -> ObjectSBH
- each name is unique
- many NC's in same address space



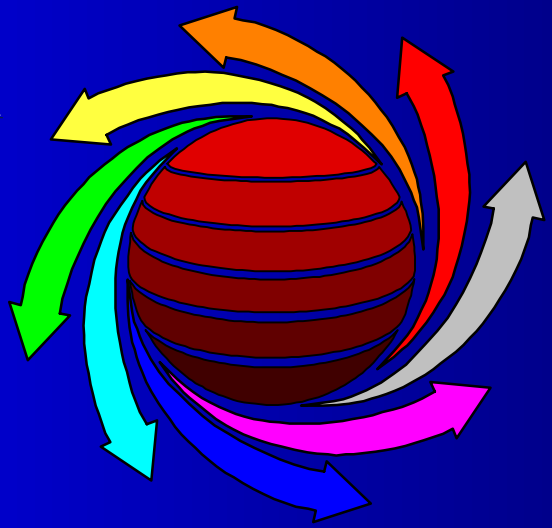
Present DL


dl
dl
dl
users
hector
services
hassan
payment agents
Webcrawler
Dialog
Folio
hassan
payment
agent
User
Service
cousins


Present DI
thames
whale
snowcrash
dl
dl
dl
thames
db
hector
users
services
washington
hassan
Webcrawler
payment agents
Dialog
Folio
hassan
payment
agent
User
Service
cousins
db2
thames

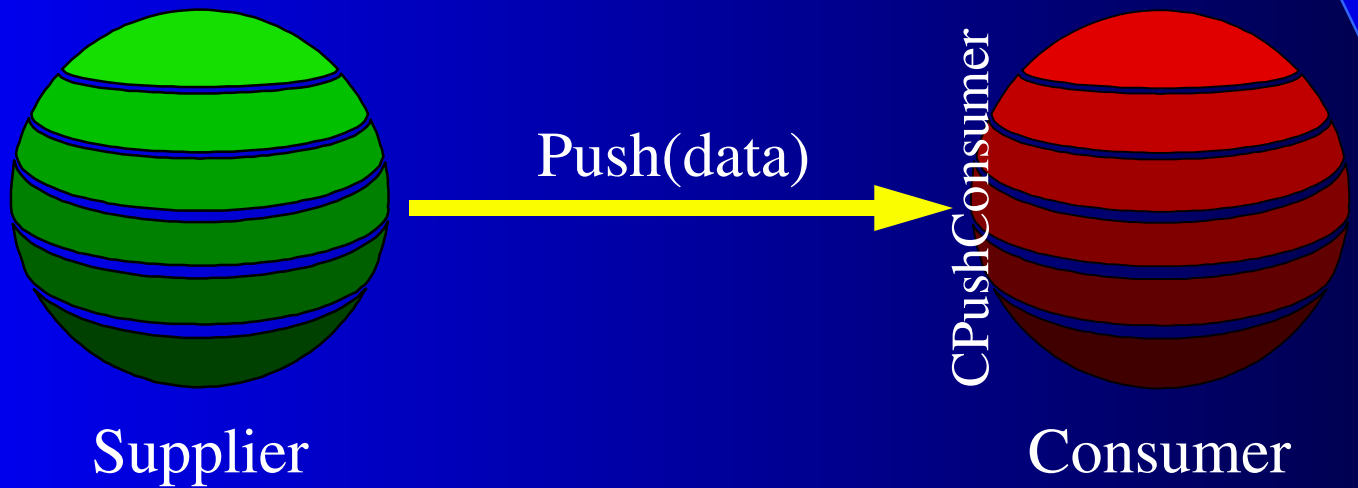
Event Services

- Decouple communcation between objects
- Suppliers - produce event data
- Consumers - process event data
- Pull model
- Push model



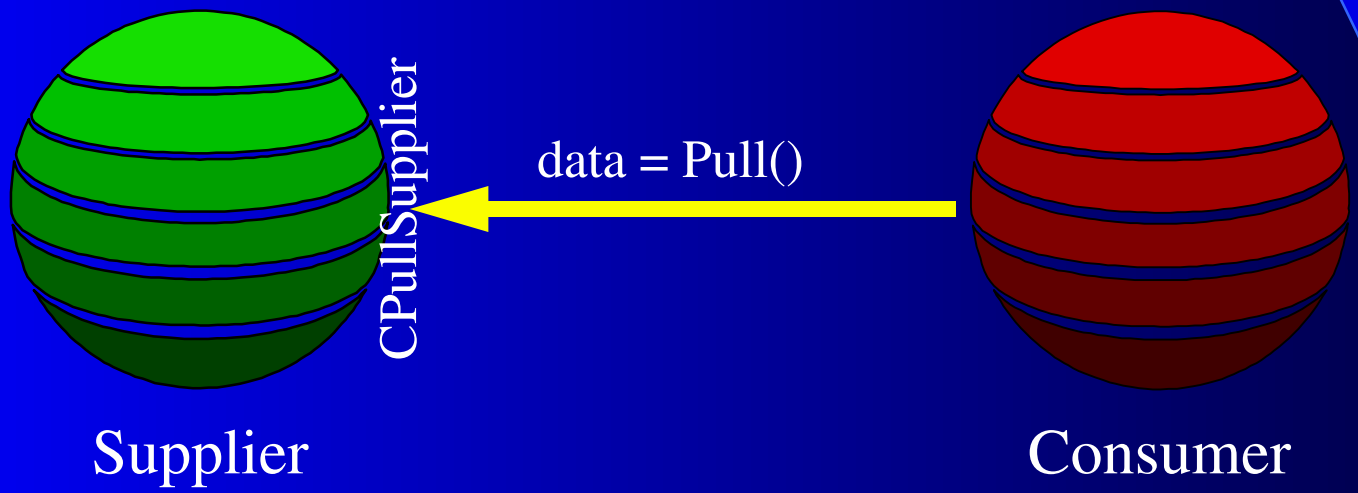
Push Model

- Suppliers invoke the ‘Push’ method on the consumer.
  - can be asynchronous

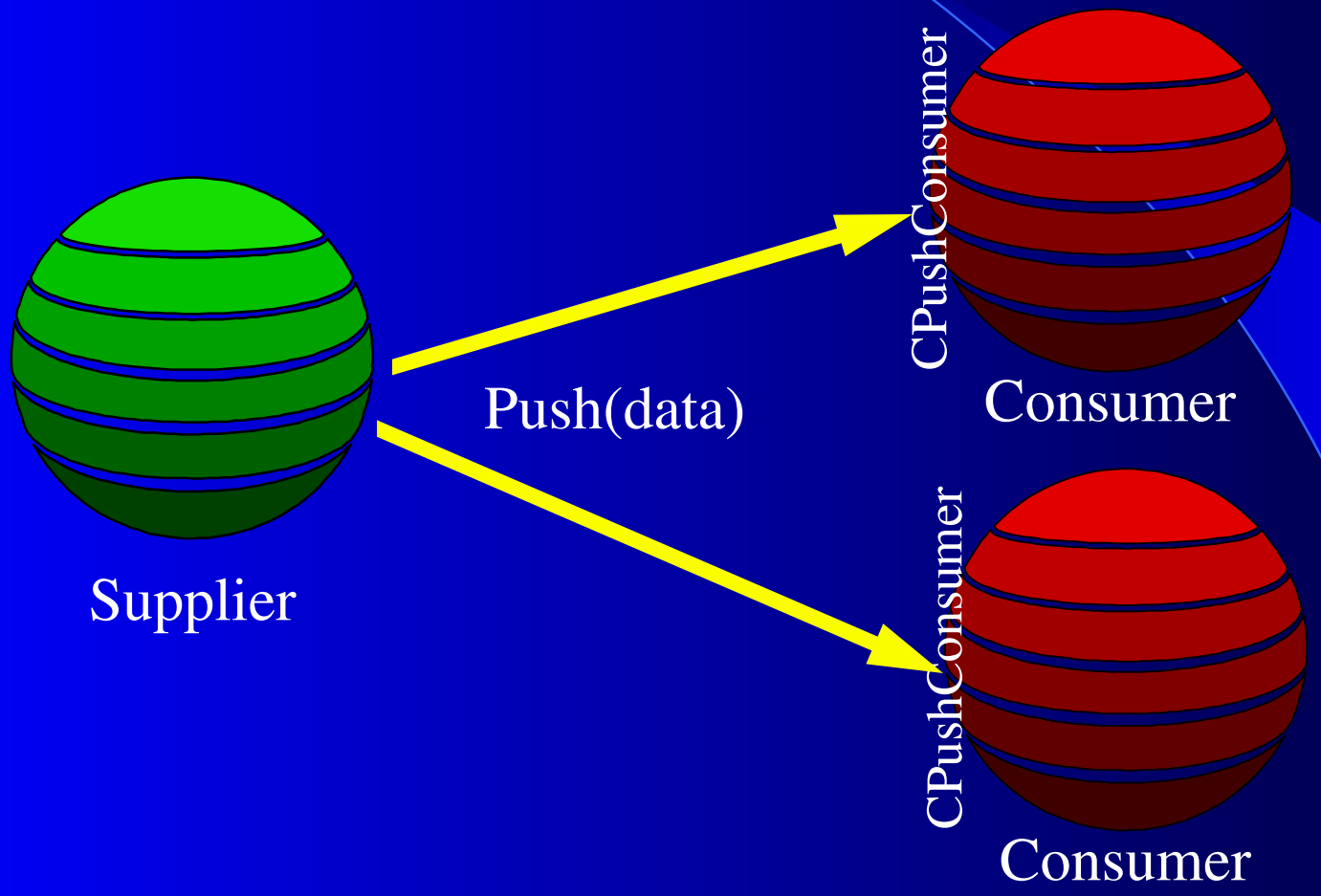


Pull Model

- Consumers invoke the ‘Pull’ method on the suppliers.
  - TryPull

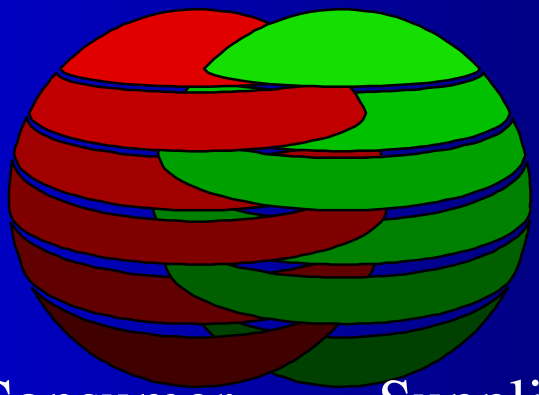


Multiple Consumers

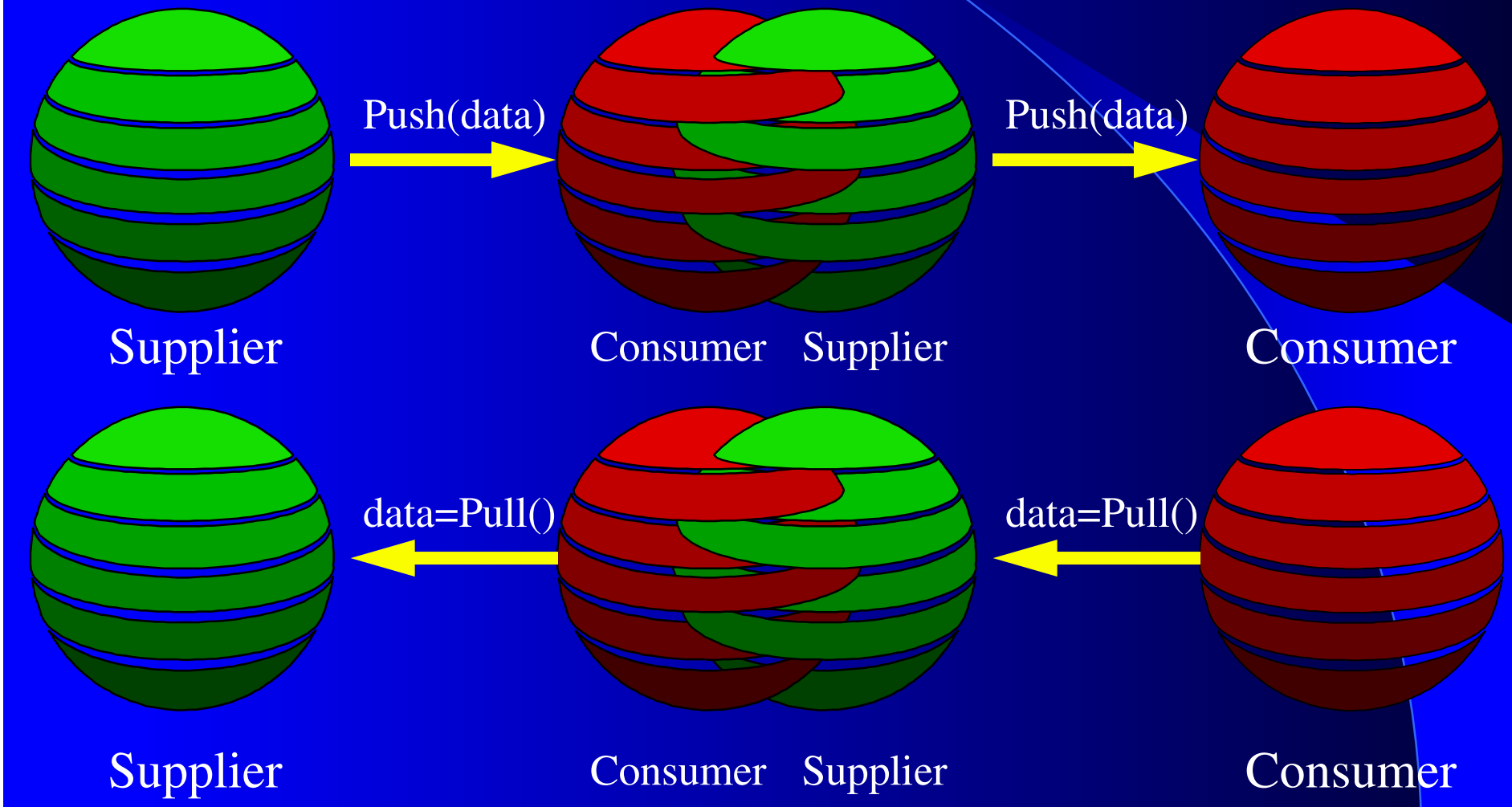


Event Channels

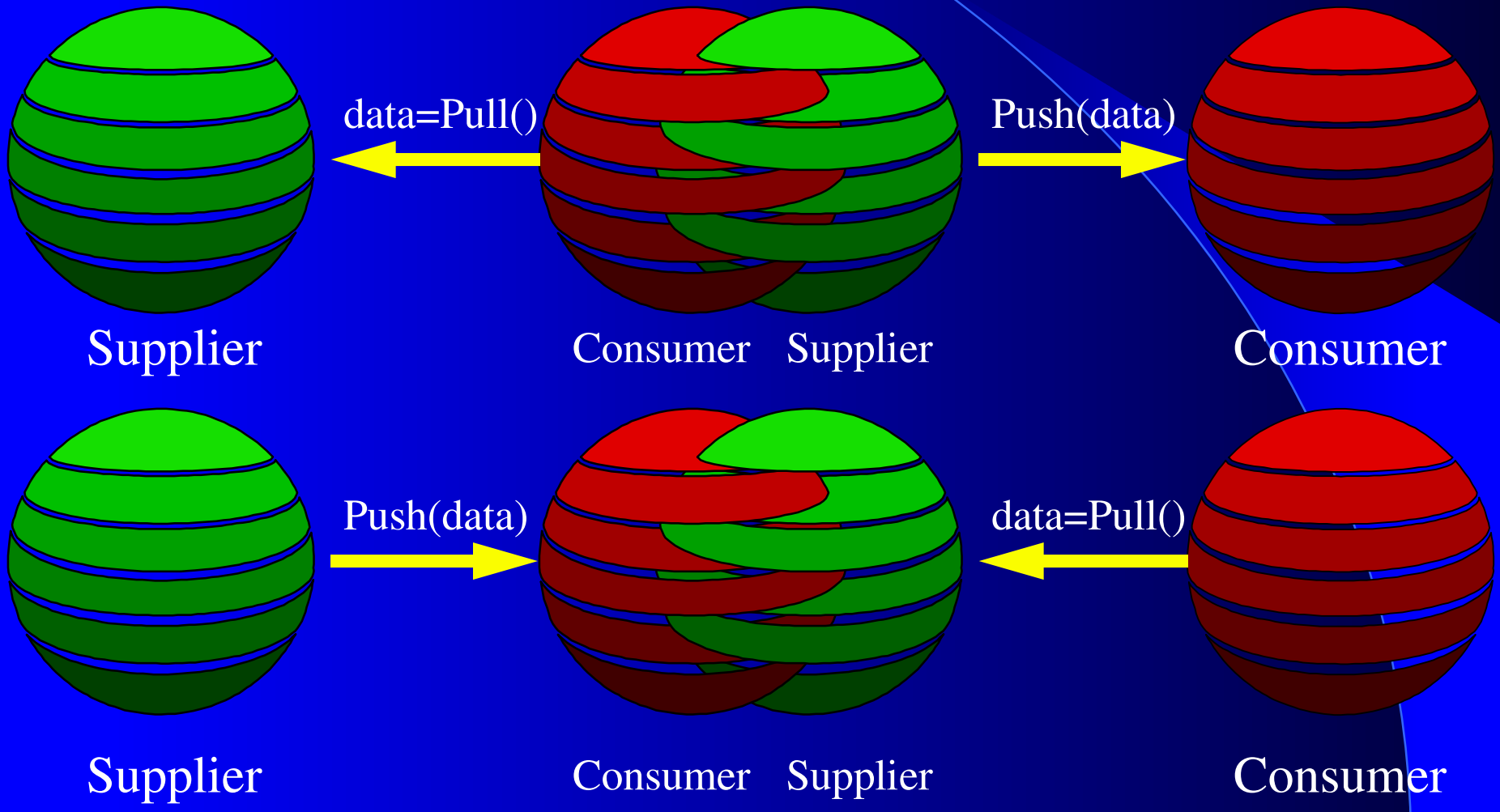
- Decouples the supplier from the consumer
- Acts as a consumer and supplier
- Asynchronous communciation
- Multiplexing
- Time delay
- Anonymous
- Proxy
- Filter



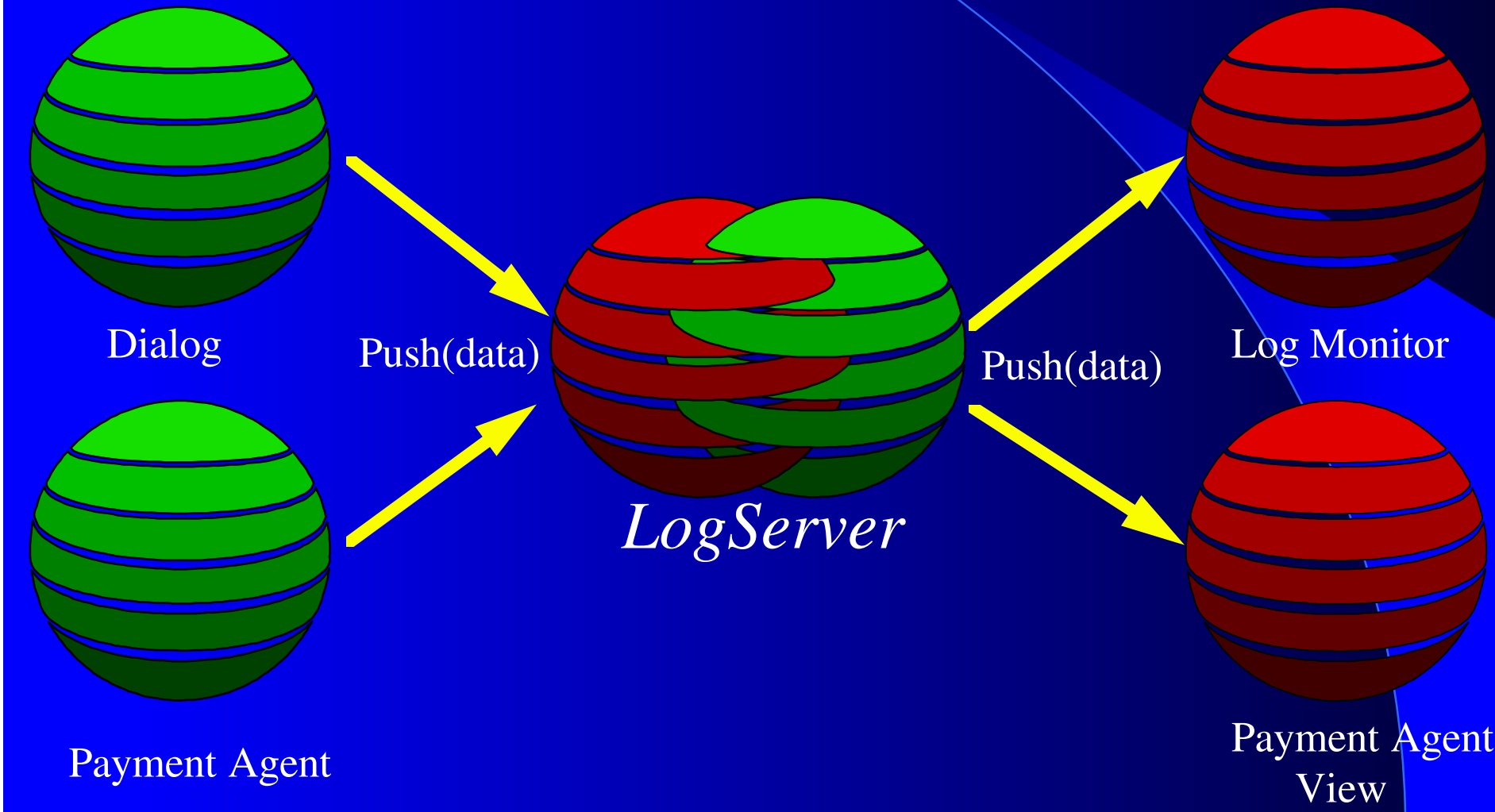
Push and Pull Event Channels



Mixed Event Channels



Example: Logging Service

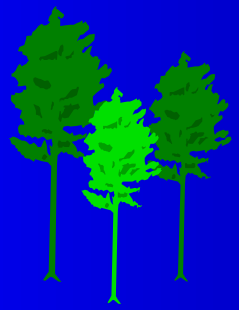


Event Notes

- Quality of service
  - best effort/guaranteed
- Security
- Registration
- Performance
- Optimizations
  - batching

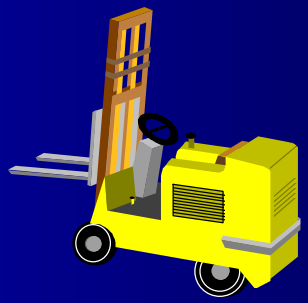
Life Cycle Services

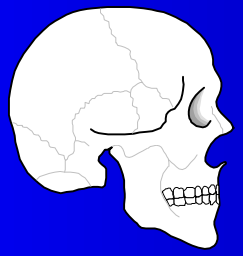
Create



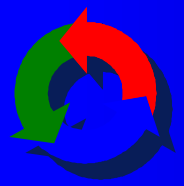
Copy

Move





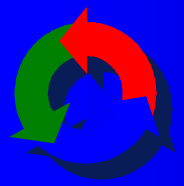
Remove

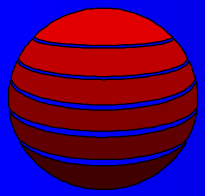


Create

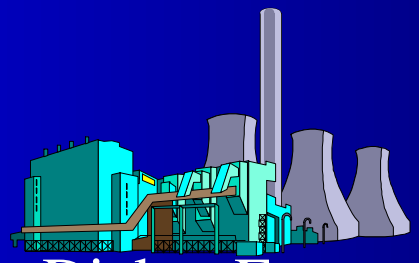
- Clients ask ‘factory’ objects to create objects
  - Type
    - Interface
    - Implementation
  - Criteria
    - State
    - Parameters

Factories



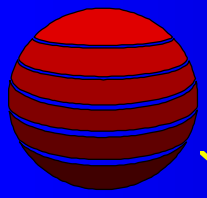


Client



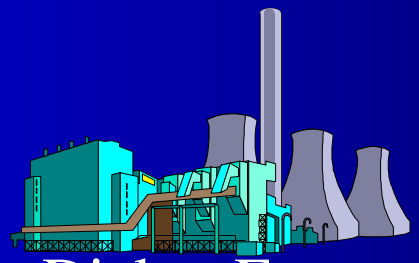
Dialog Factory

Factories



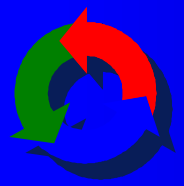
Client

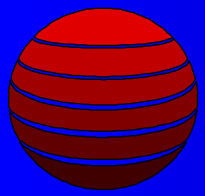
Create(dialog)



Dialog Factory

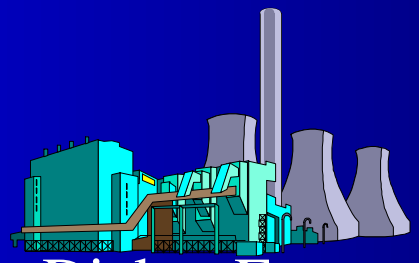
Factories





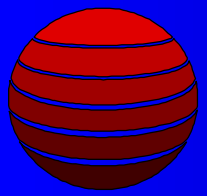
Client

Dialog



Dialog Factory

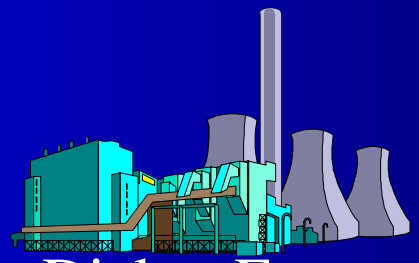
Factories



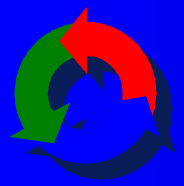
Client

Search()

Dialog

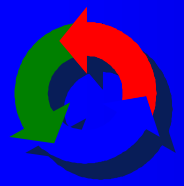


Dialog Factory



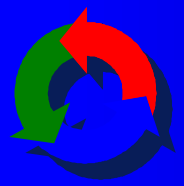
CLifeCycle

- Supports Copy, Move, and Remove
- Most DL Objects should inherit CLifeCycle

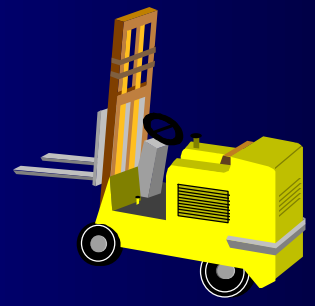


Copy

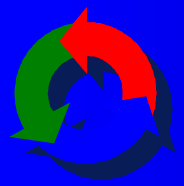
- Client ask an object to copy itself.
  - Object of same type
  - Criteria
    - State
    - Parameters
- Algorithm
  - Creates an object of same type
  - Transfer state information
  - Returns reference to new object



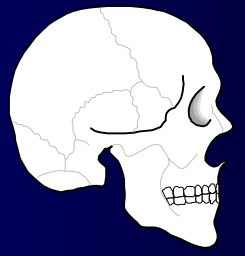
Move



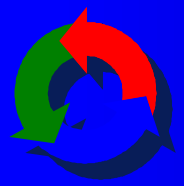
- Client ask an object to move itself.
  - Object of same type
  - Criteria
    - State
    - Parameters
- Algorithm
  - Creates an object of same type
  - Transfer state information
  - Destroy self
  - Returns reference to new object



Remove

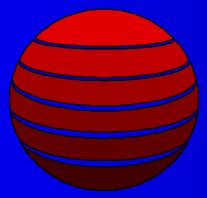


- “Instructs the object to cease to exist”
- Call ‘Remove’ method on CLifeCycle object
  - cleans up
  - notify others
  - die



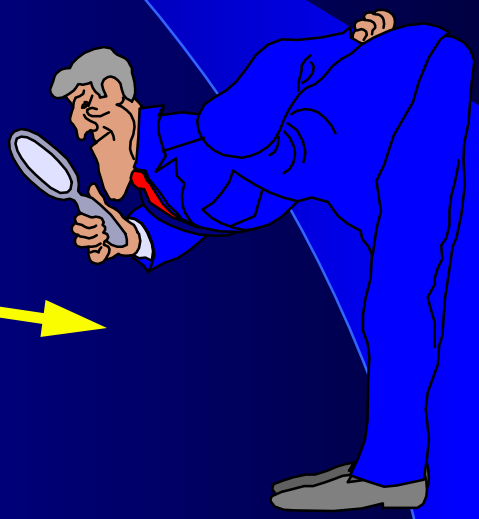
Factory Finders

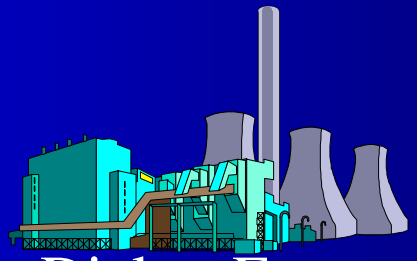
- Client asks for a factory of a certain type
- Scope to creation of new objects



Client

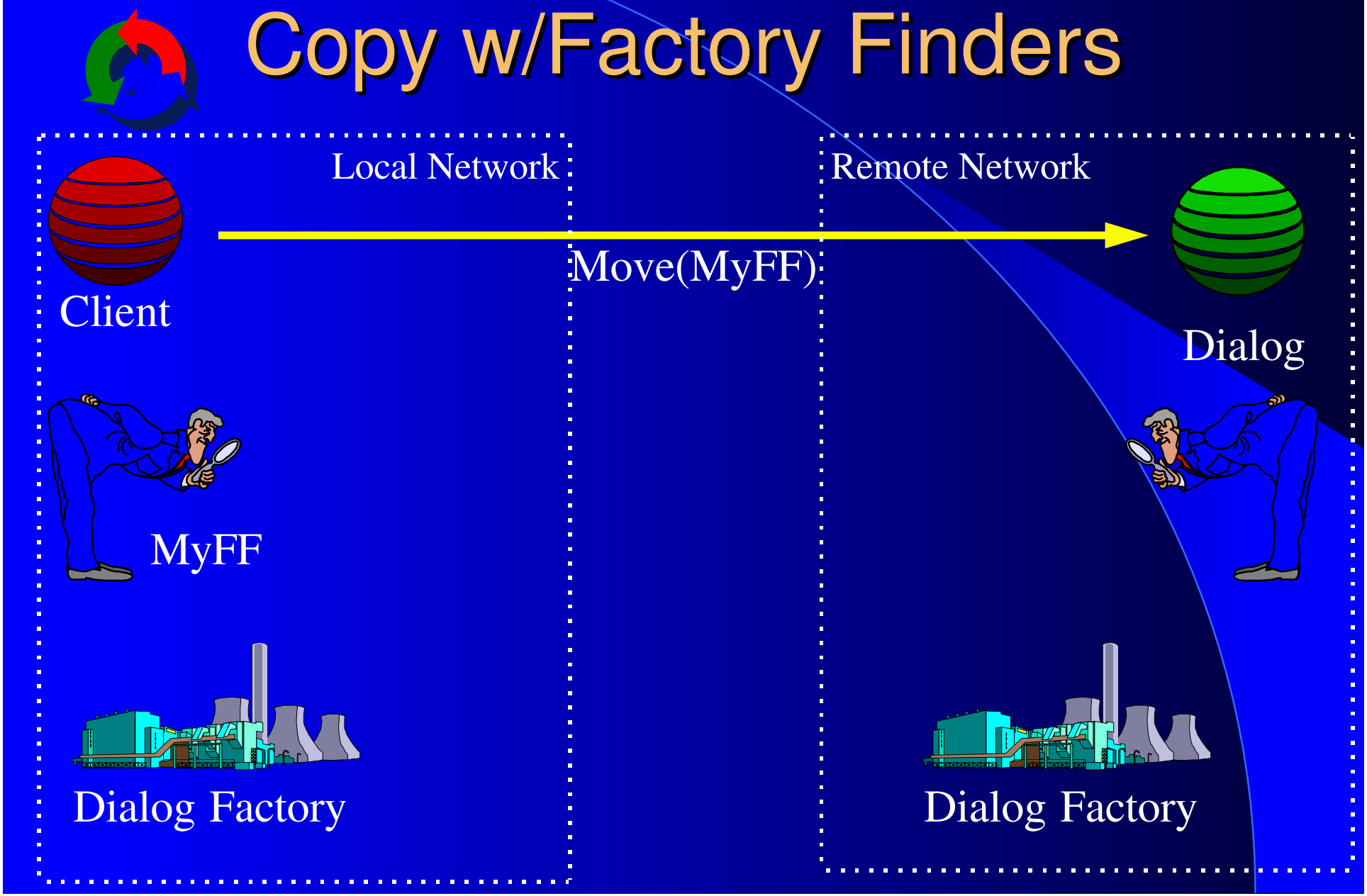
FindFactory(Dialog)



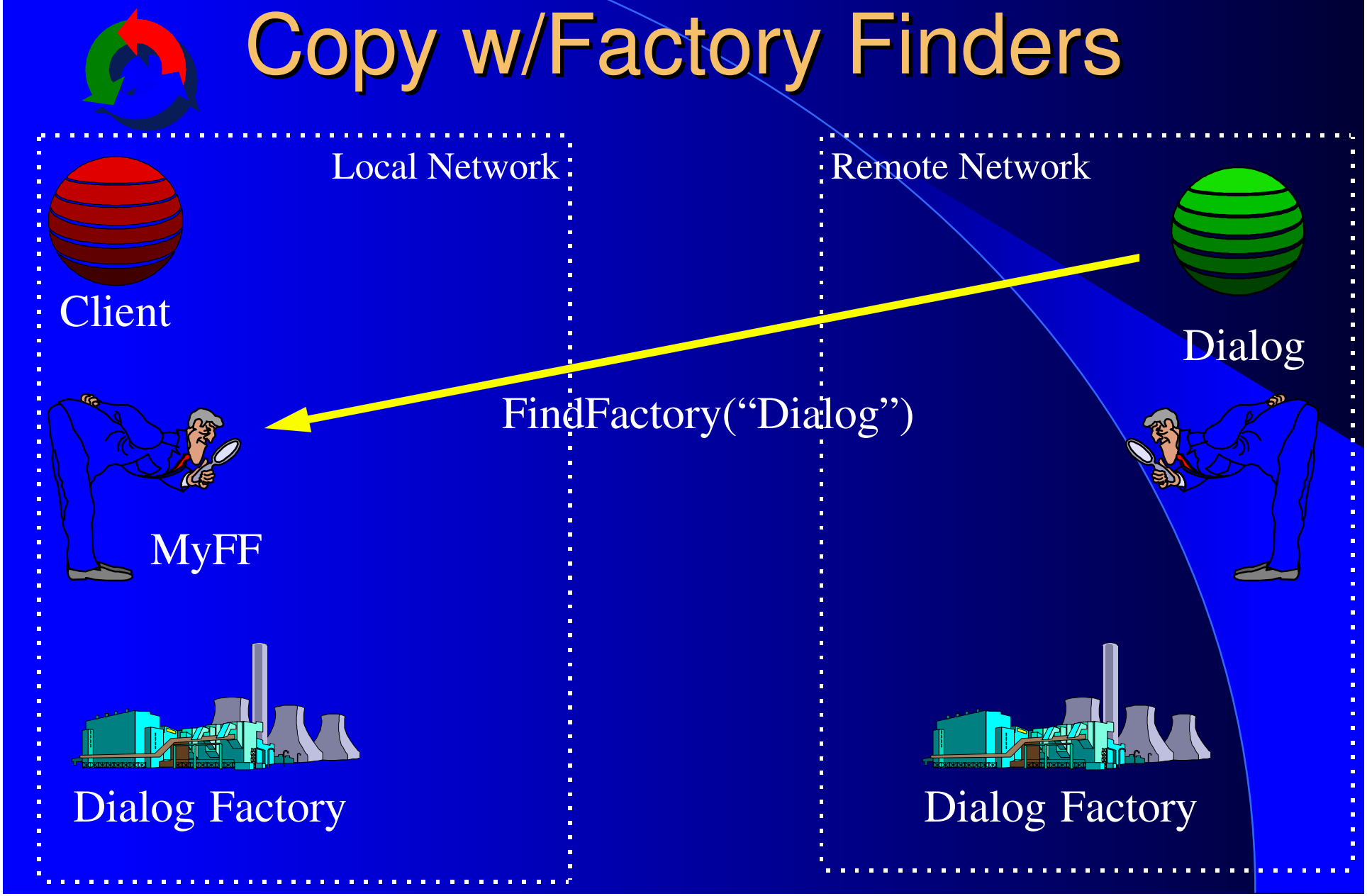


Dialog Factory

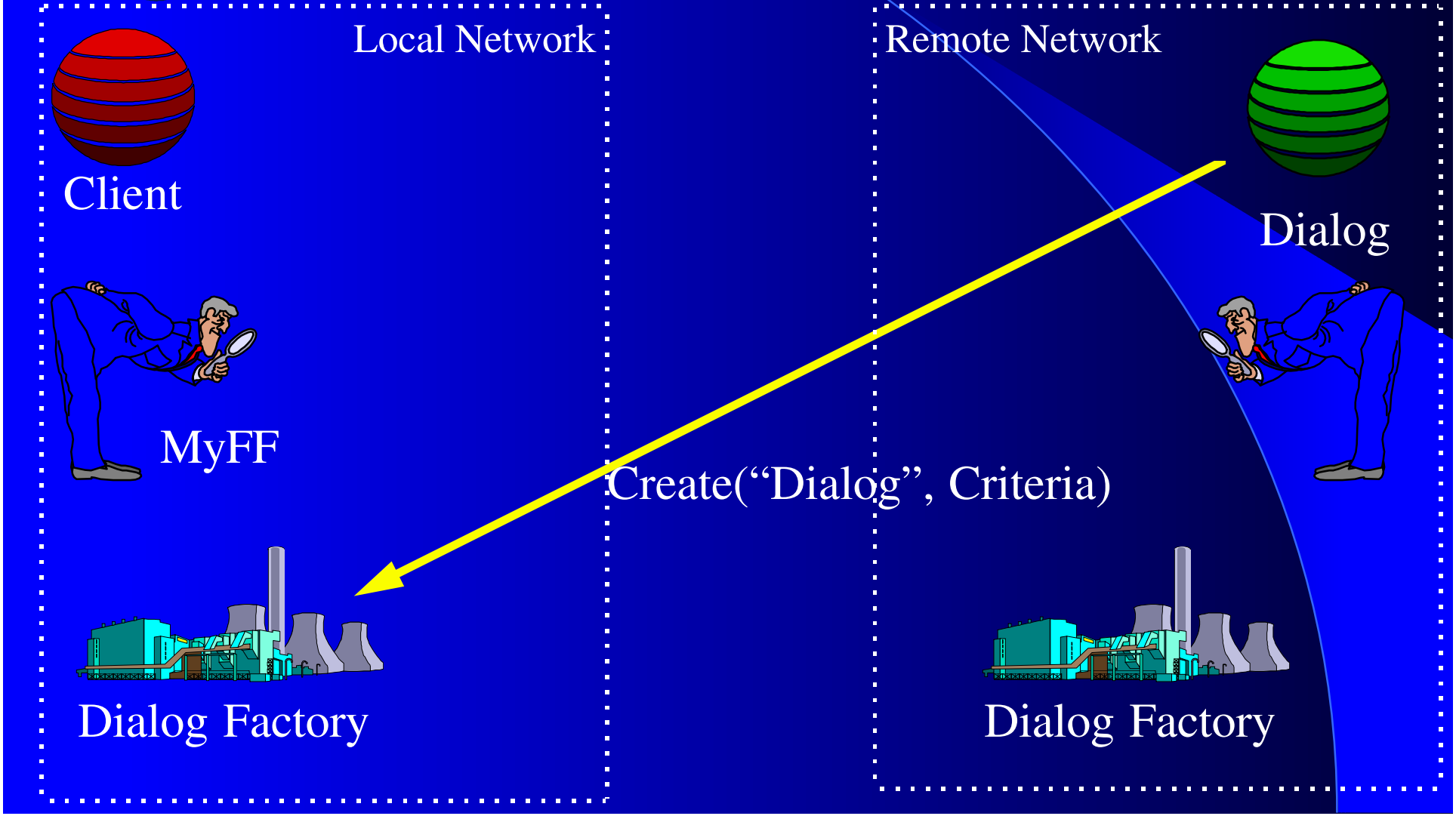
Copy w/Factory Finders



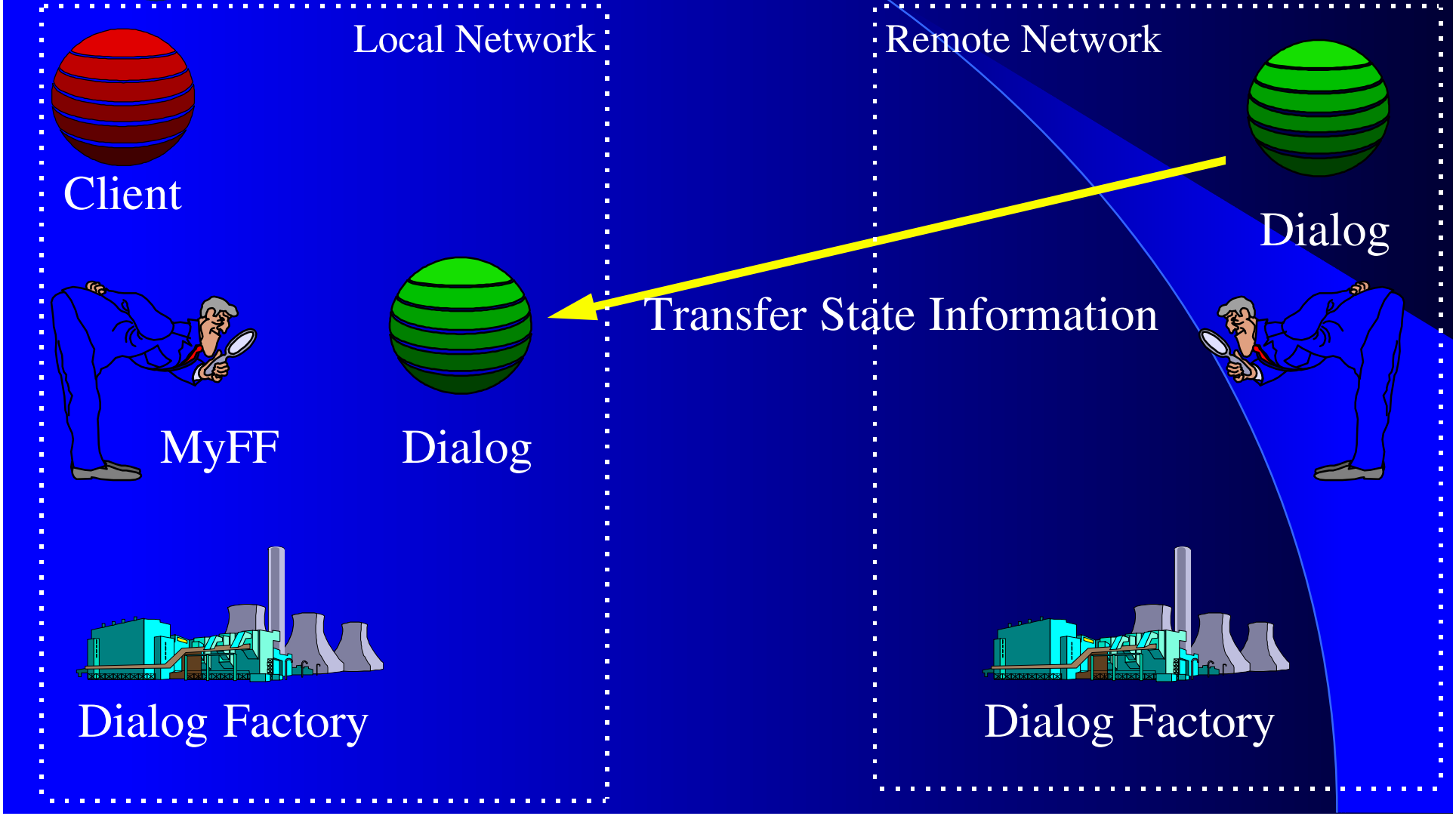
Copy w/Factory Finders



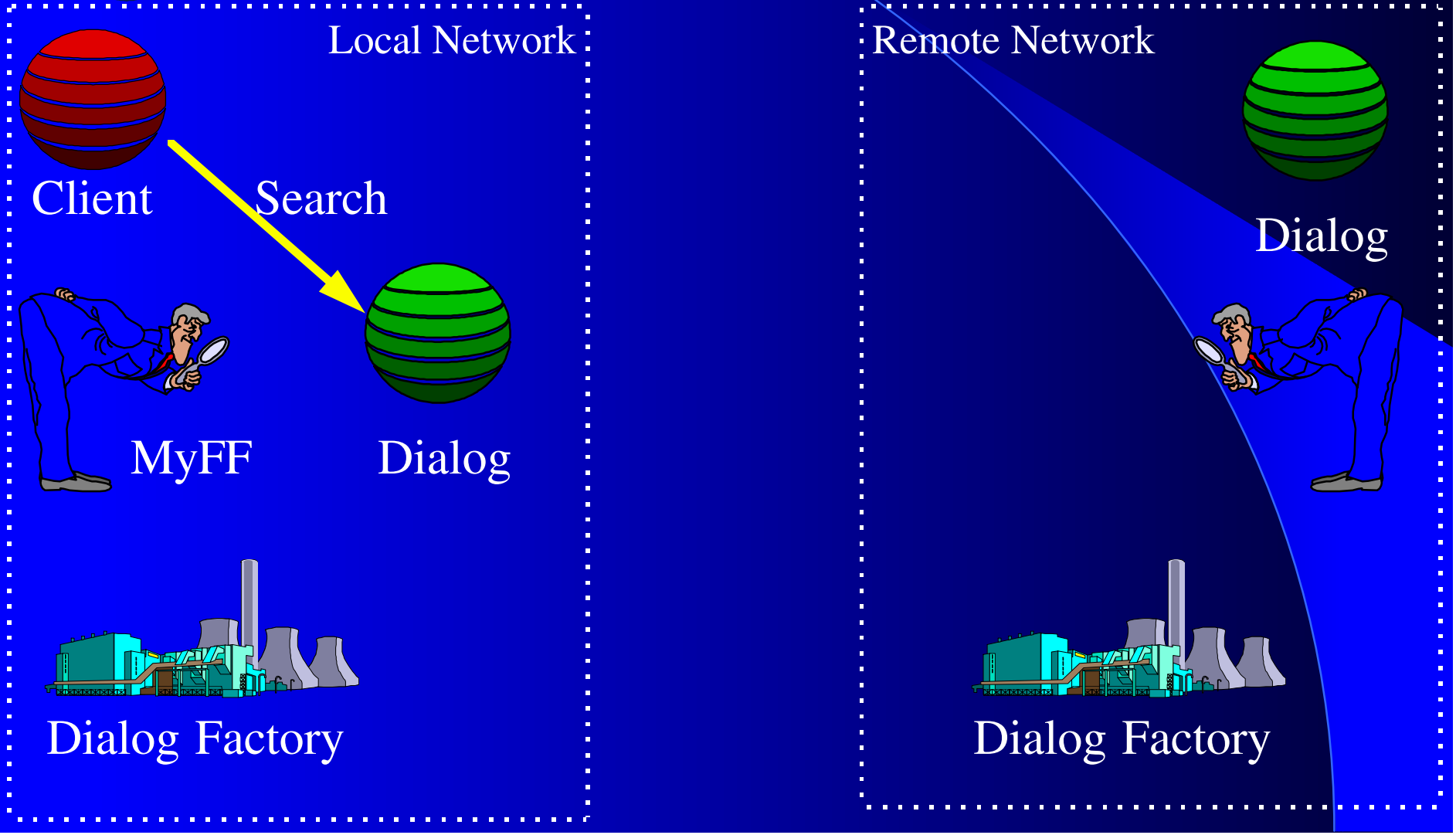
Copy w/Factory Finders

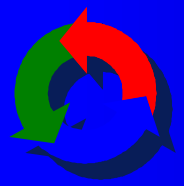


Copy w/Factory Finders



Copy w/Factory Finders





Life Cycle Notes

- Deep/Shallow Copy/Move
  - Relationship Service
    - Contains
    - References

Where to get more information

- *http://www-db.stanford.edu/~testbed*
- "Common Object Services Specification V1"
  - OMG 94-1-1, John Wiley & Sons, Inc.
- Me!