Causing Mayhem on the Internet with Virtual Environments

Manuel Oliveira

Department of Computer Science

University College London

Contents

- Motivation
- Building Blocks
- Jade
- TreacleWell
- MIM
- MUD
- Q&A

Motivation: Vision

“Hiro is approaching the Street. It is the Broadway, the Champs Elysees of the Metaverse. It is brilliantly lit boulevard that can be seen, miniaturized and backward, reflected in the lenses of his goggles. It does not really exist. But right now, millions of people are walking up and down it.”

“Developers can build their own small streets feeding off the main one. They can build buildings, parks, signs, as well as things that do not exist in Reality, such as vast hovering overhead light shows, special neighbourhoods where the rules of three-dimensional spacetime are ignored, and free-combat zones where people can go hunt and kill each other.”

Motivation: Virtual Environments?

- **Virtual Environments are:**
  - Small budgets – no cool factor
  - Genesis (2-4 years) then on-going
  - Dynamic milestones
  - Limited to few well behaved users
  - Rich variety of architectures

Motivation: Online Games?

- **Online games are:**
  - Virtual Environments (VE) with fat budgets
  - Built on graphic research ideas (i.e: Quake)
  - Stressed by players
  - Non-evolvable engines
  - Based on client/server architectures

Motivation: Bane of current VE

- Monolithic systems
- Functional limitation - Inflexibility
- Where is scalability?
- Network continues to be “black magic”
- Steep learning curve
- Laws of online game development

Motivation: Siggraph 2000 Panel

- Perception of components
- Re-inventing the wheel syndrome → VE developers
- Not invented here syndrome → System developers
- Position of Industry to open source

Motivation: What do we want?

Systems that support:

- Persistency → cyber communities
- Runtime Extensibility → System Evolution
- Interoperability → content and code
- Global Infrastructure → scalable

Motivation: What do we want?

Systems that support:

- Persistency → cyber communities
- Runtime Extensibility → System Evolution
- Interoperability → content and code
- Global Infrastructure → scalable

IT IS ALL ABOUT PEOPLE

and most prefer fishing

Motivation: Where are we?

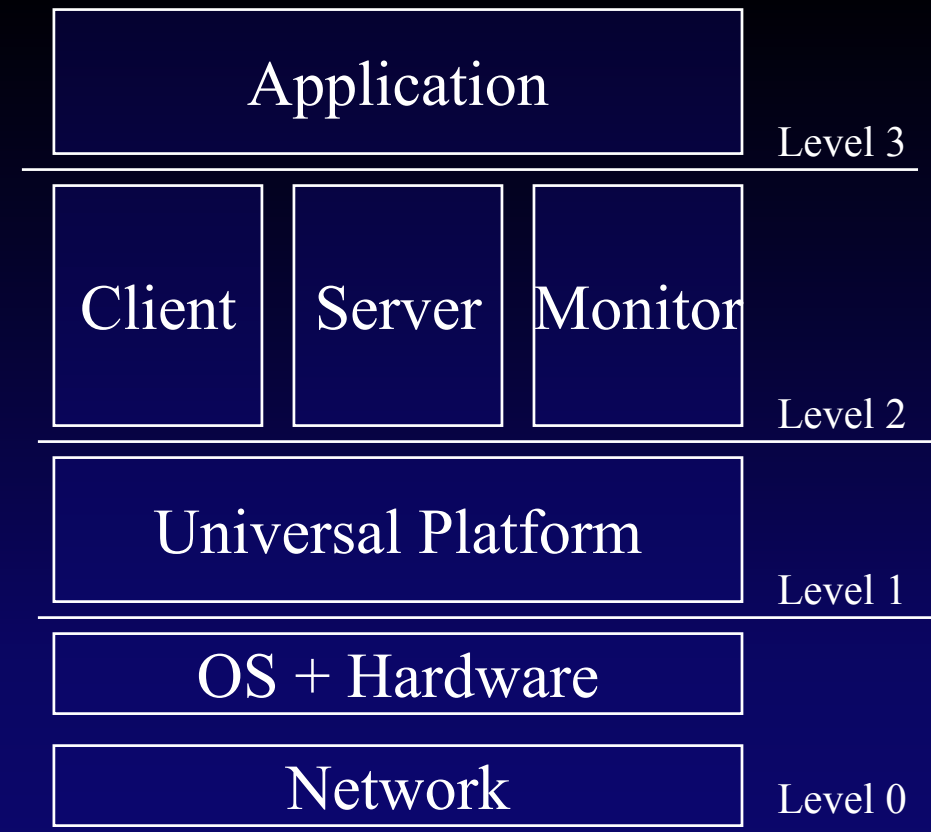
- The birth of cyberculture
- Lots of good systems and games
- No experience exchange
- New idea → new system
- Lots of complains
- Lots of frameworks
- NO PROGRESS!!!

Motivation: What does this mean?

- We have an understanding of issues involved
- Lack communication
- There is no “right” way
- Like reinventing the wheel
- Insufficient maturity to standardize
- Know-it-all systems

Building Blocks: Component Framework

- **VRTP Framework:**
  - Macro components – every other framework fits
  - The lower the level the more flexibility there is but the less the framework does for you
  - At the highest level, you just start the application



Building Blocks: UP Requirements

- **Universal Platform**
  - Independent of system resources + cross-platform
  - Retrieve resources/components
  - Some network capabilities
  - Component management
  - Global namespace for components

Building Blocks: So what is Mayhem?

- **A set of well defined frameworks**
- **A reference implementation of the frameworks and specific extensions**
- **A sample application of a game that supports:**
  - Heterogeneity
  - Scalability
  - Interoperability
  - Runtime extensibility

Building Blocks: Mayhem Operation

- Launch JADE with options
- Service bootstrap in XML
- Connect Server and user validation
- Capability negotiation
- Download XML description of system (ie: TW, MIM, MUD, etc)
- Enter a world
- Download the content (ie: X3D)
- Announce user
- Update and exchange content

Building Blocks: Mayhem frameworks

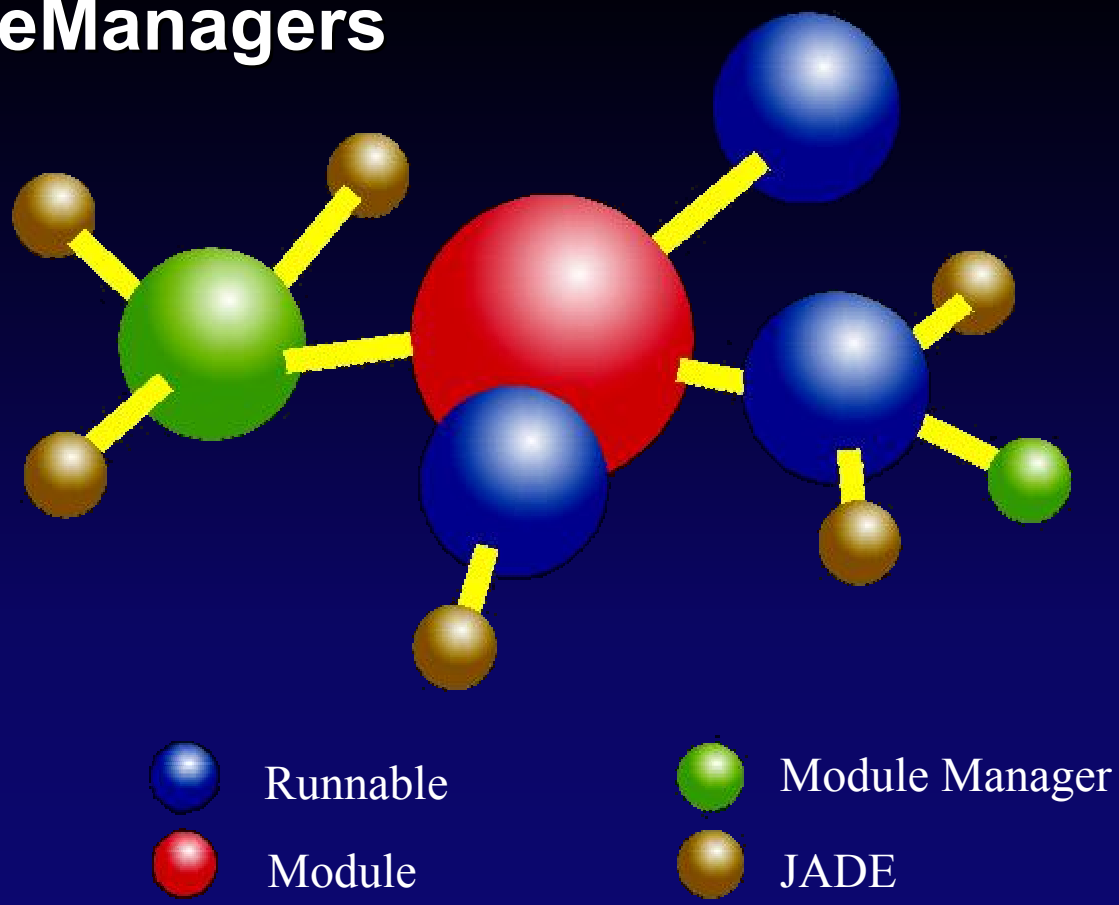
- **So what are the Mayhem frameworks:**
  - Java Adaptive Dynamic Environment (JADE)
  - TreacleWell
  - Meta Interest Management (MIM)
  - Meta Unified Datamodel (MUD)

JADE: Java Adaptive Dynamic Environment

- **In a nutshell:**
  - It is a low-level framework for component management
  - Reference implementation
  - Framework more important than implementation
  - Uses Java

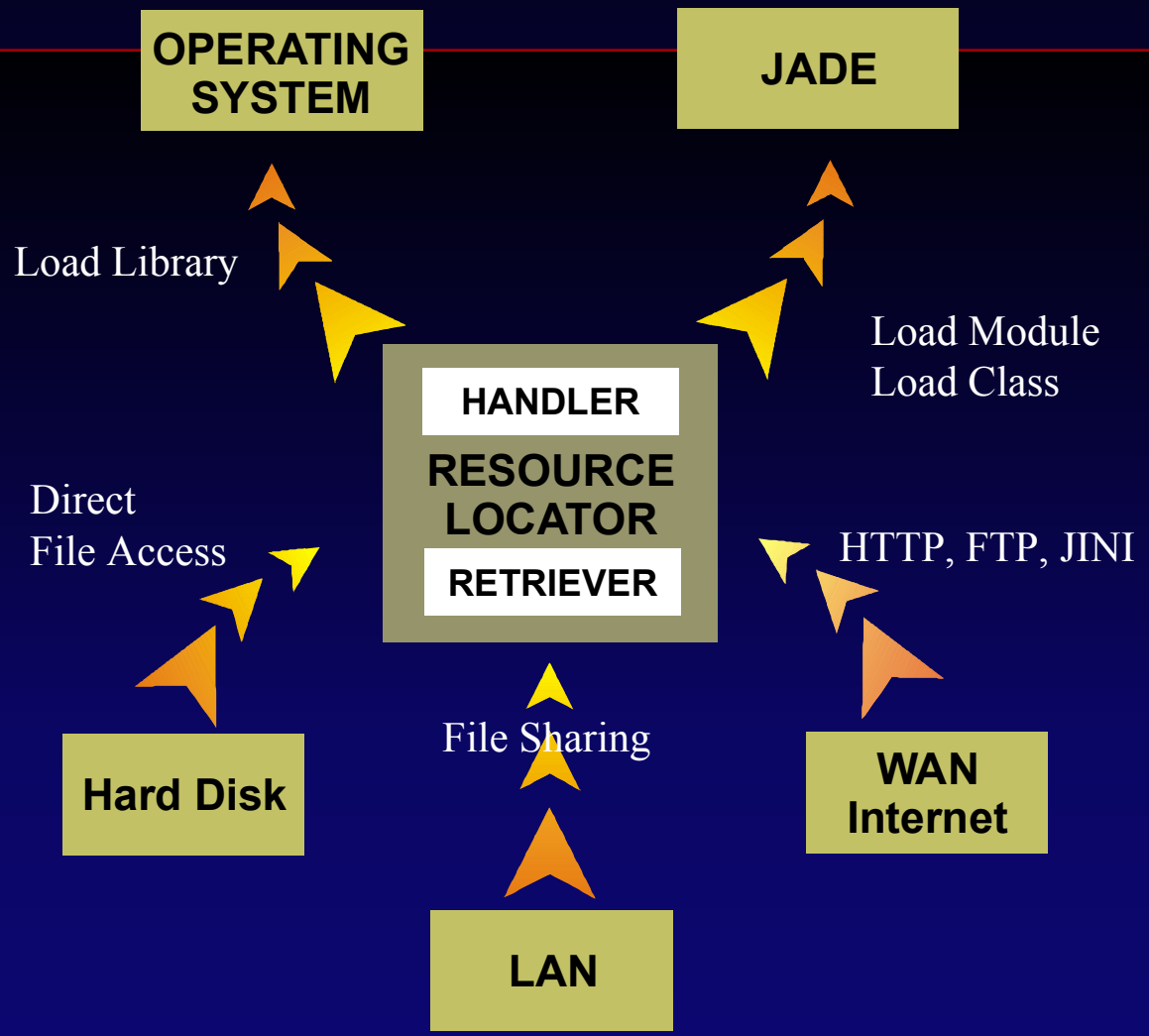
JADE: Overview

- **Containers : ModuleManagers**
  - Load/Unload
  - Reload + policy
  - Find/Retrieve
- **Modules**
  - Initialize
  - Activate
  - Deactivate
  - Shutdown



JADE

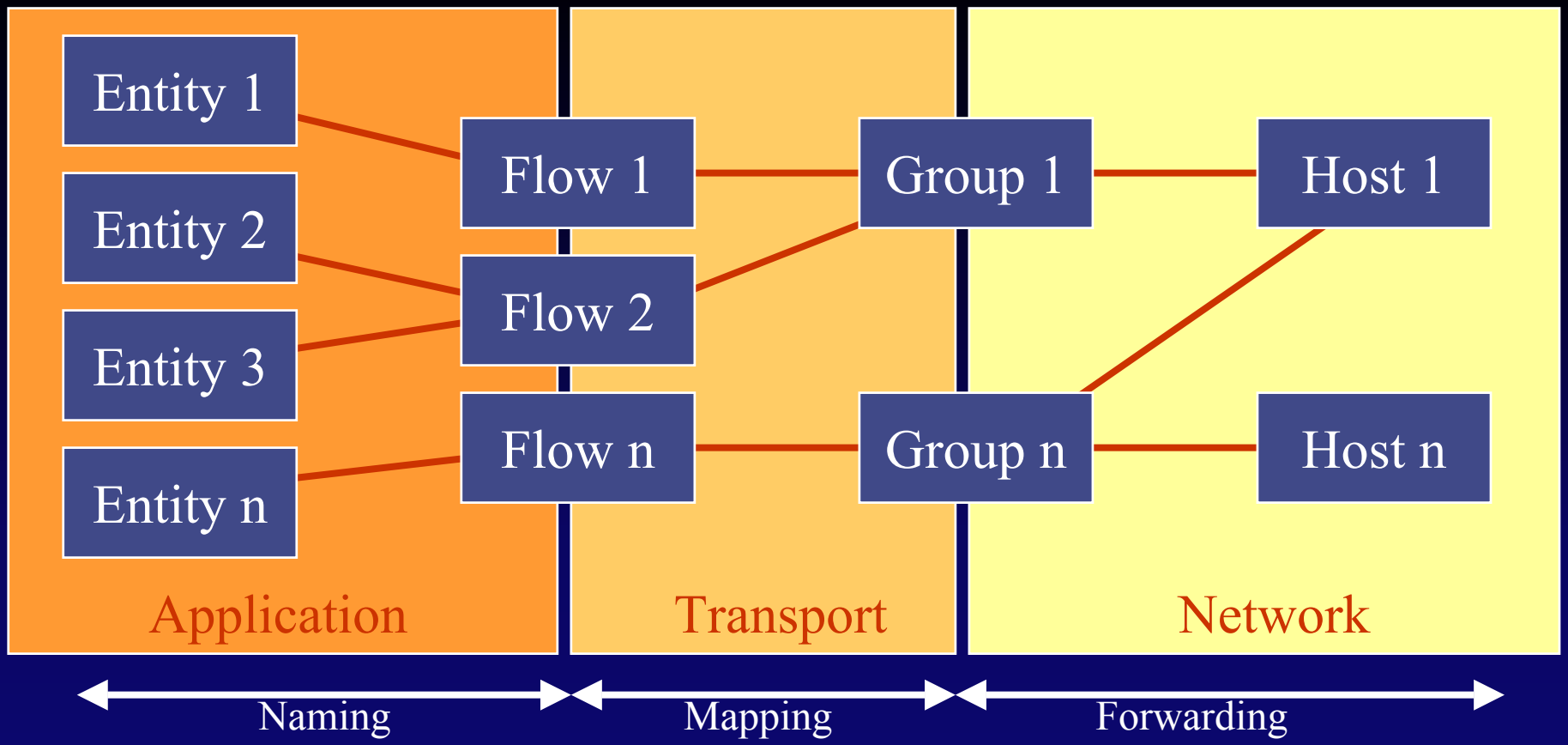
- **Resource retrieval**
  - Any source
  - Any sink
  - Extensible
  - Name service (Jini, CORBA, LDAP, etc)



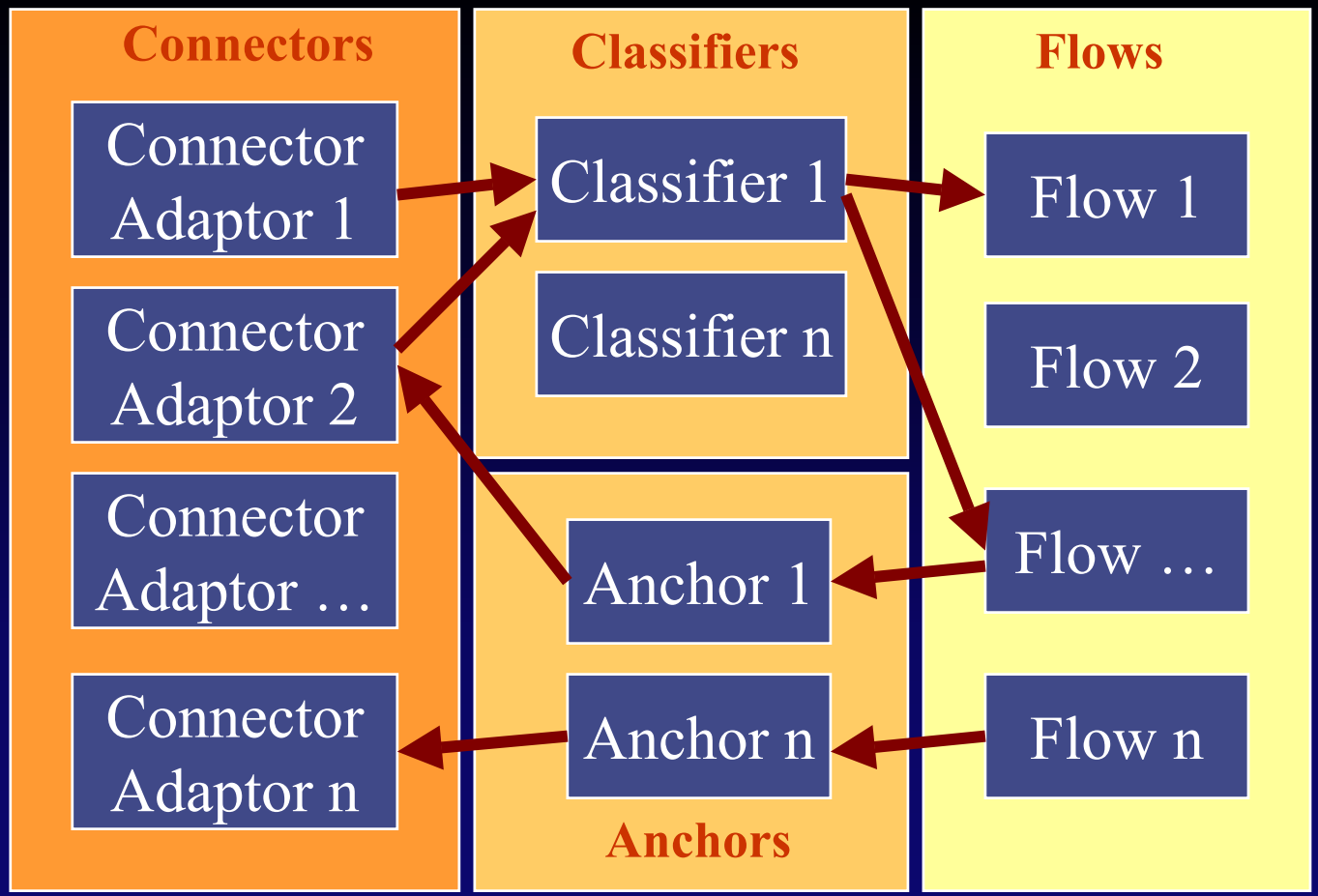
TreacleWell: What is it?

- The UP network Module
- Based on Application Level Framing and Micro-protocols concepts
- Unified namespace
- Abstraction to sockets and packets
- Provides component protocols (fragmentation, reliability, time synchronization, etc)

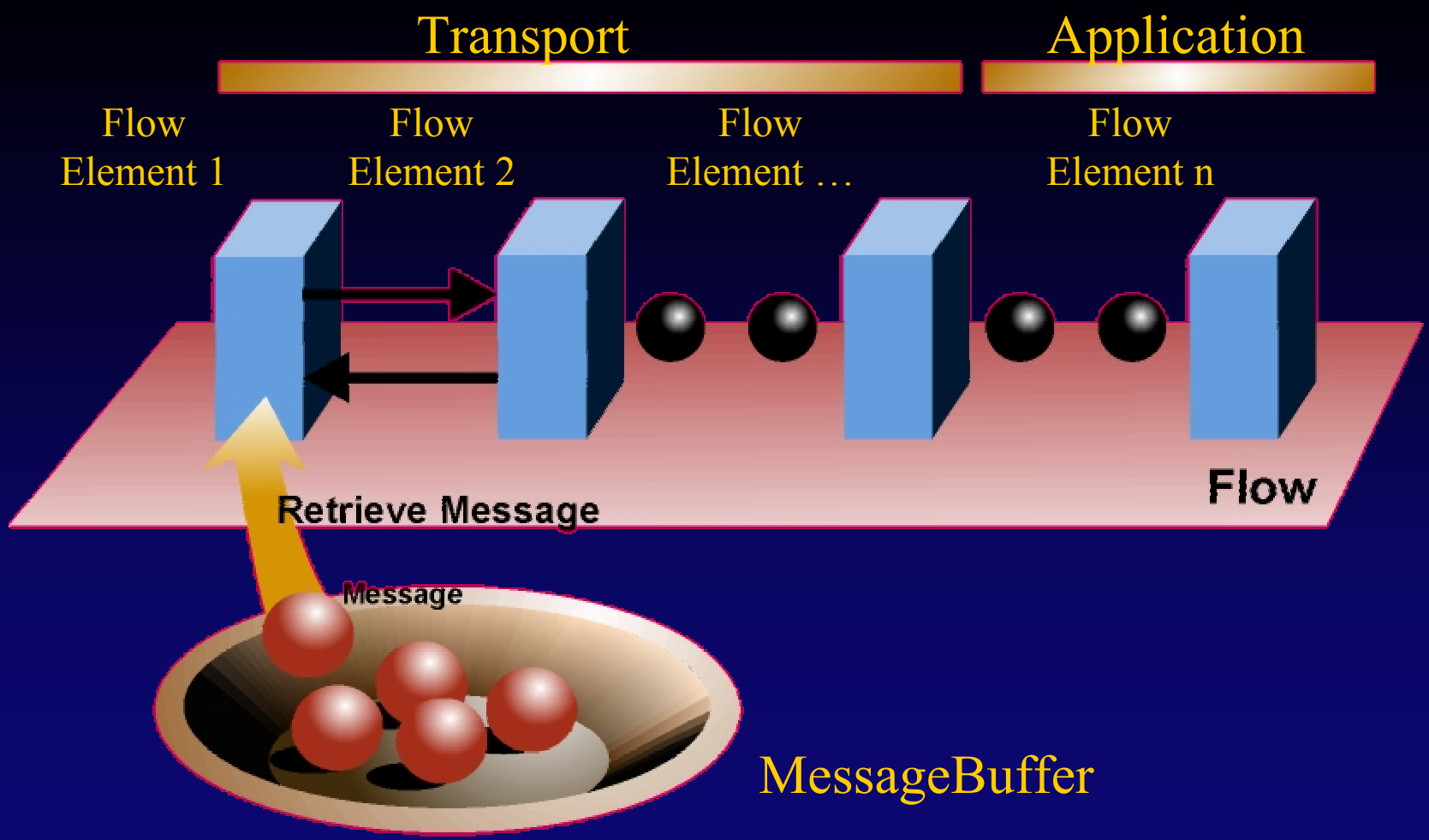
TreacleWell: The Global Picture



TreacleWell: The Framework



TreacleWell: Flow Structure



TreacleWell: Slow as molasses?

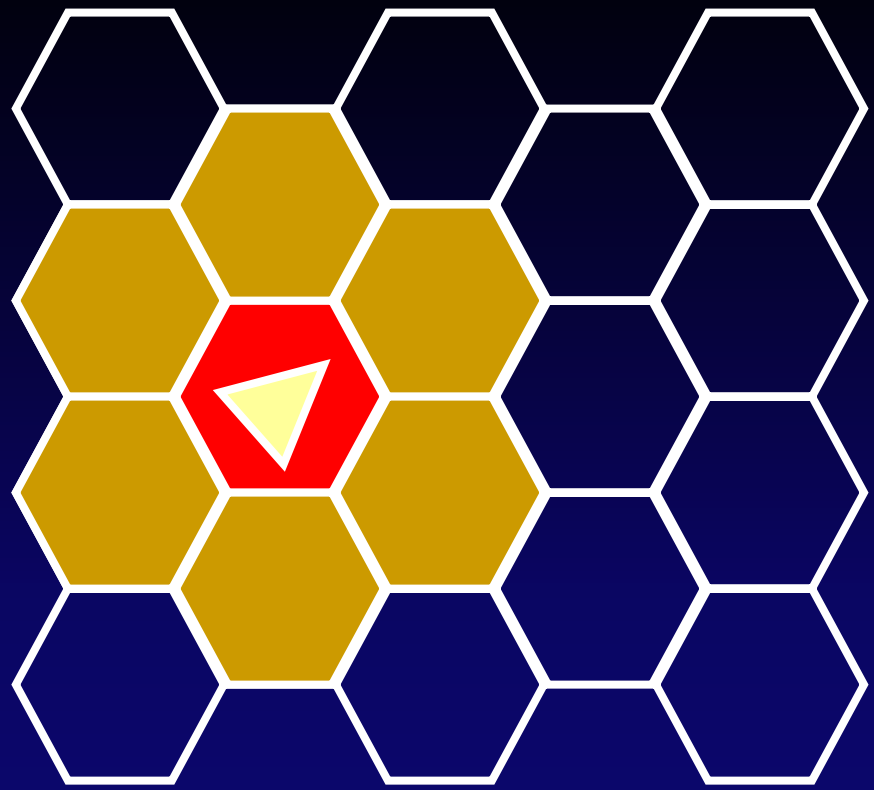
- **UDP Connector:**
  - Send Messages: 13500 messages/s
  - Receive Messages: 3825 messages/s
- **Flows:**
  - Flow A Send: 7745 messages/s
  - Flow B Send: 13297 messages/s

Flow A = position/string/time

Flow B = string

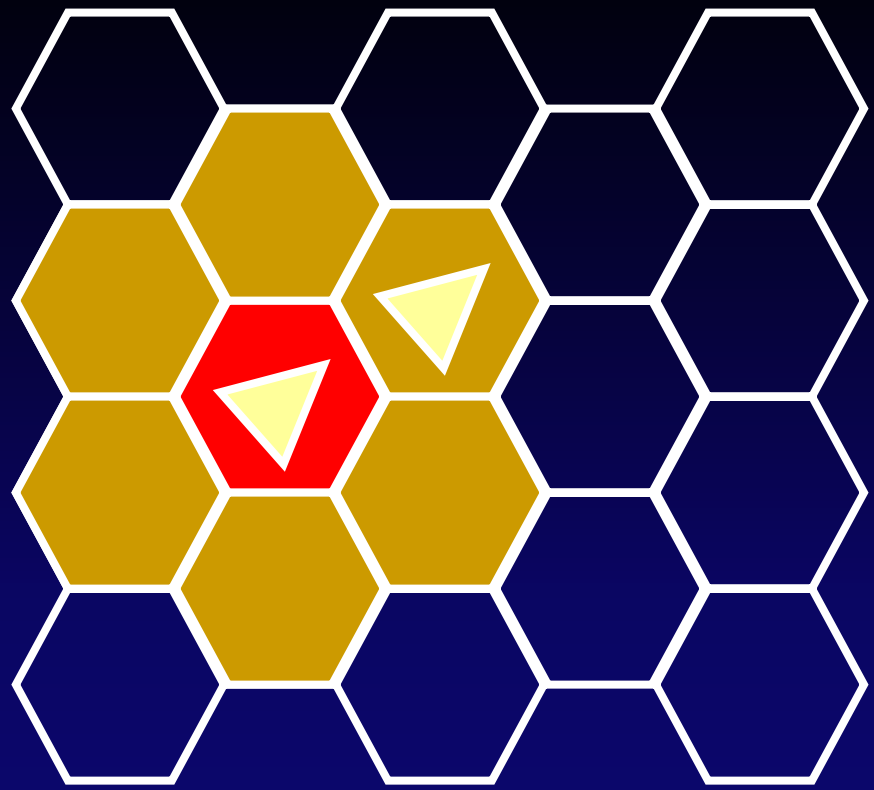
MIM: AOIM types

- Spatial



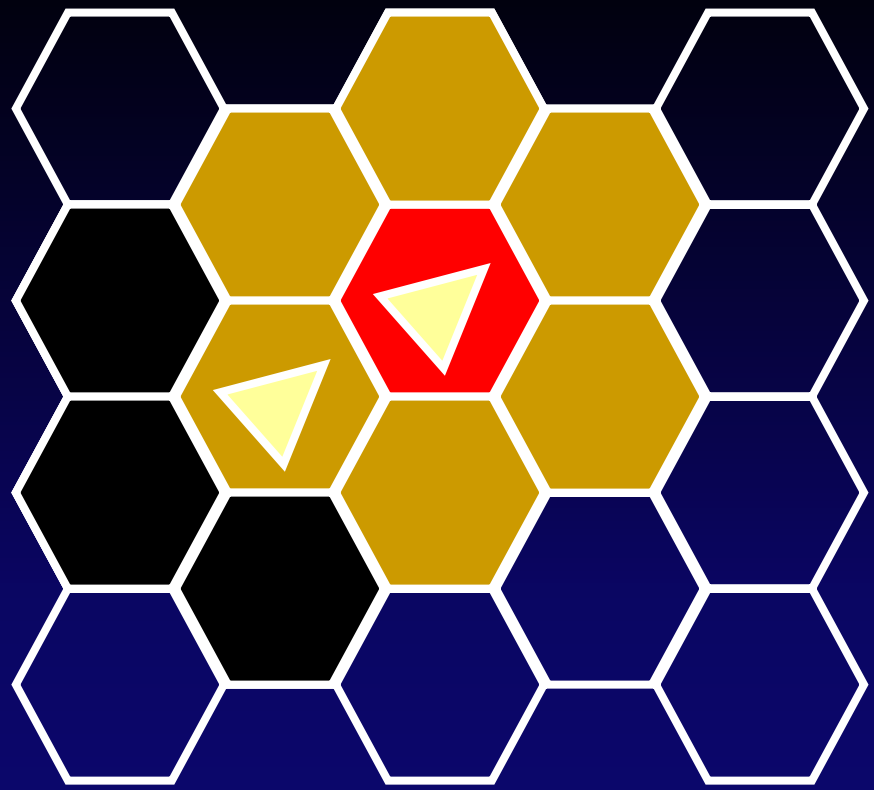
MIM: AOIM types

- Spatial



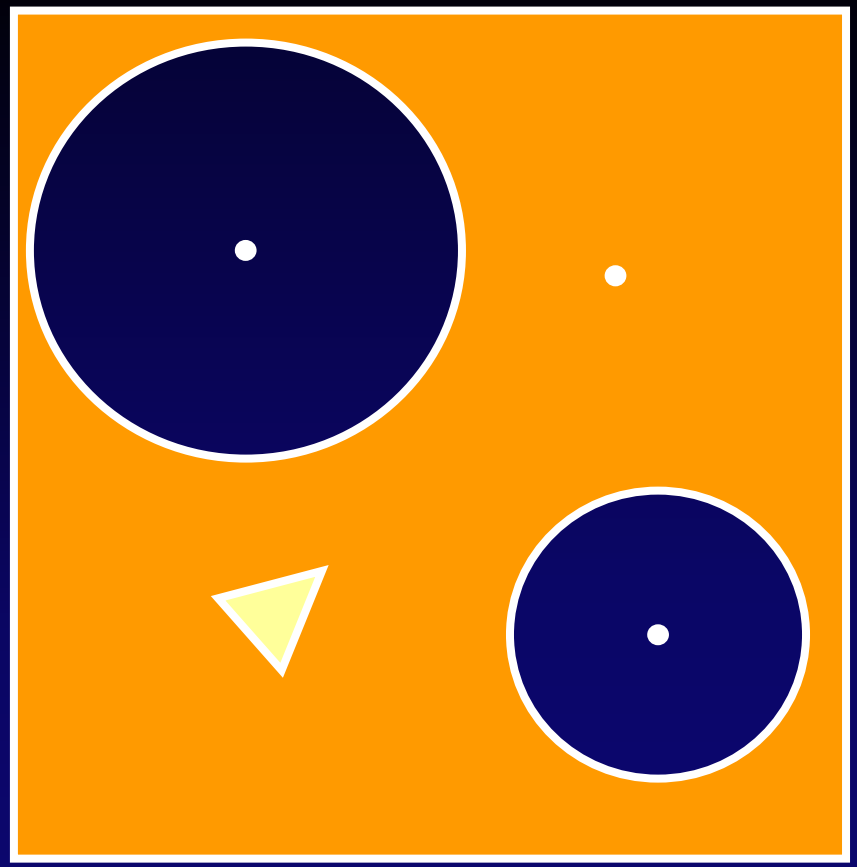
MIM: AOIM types

- Spatial



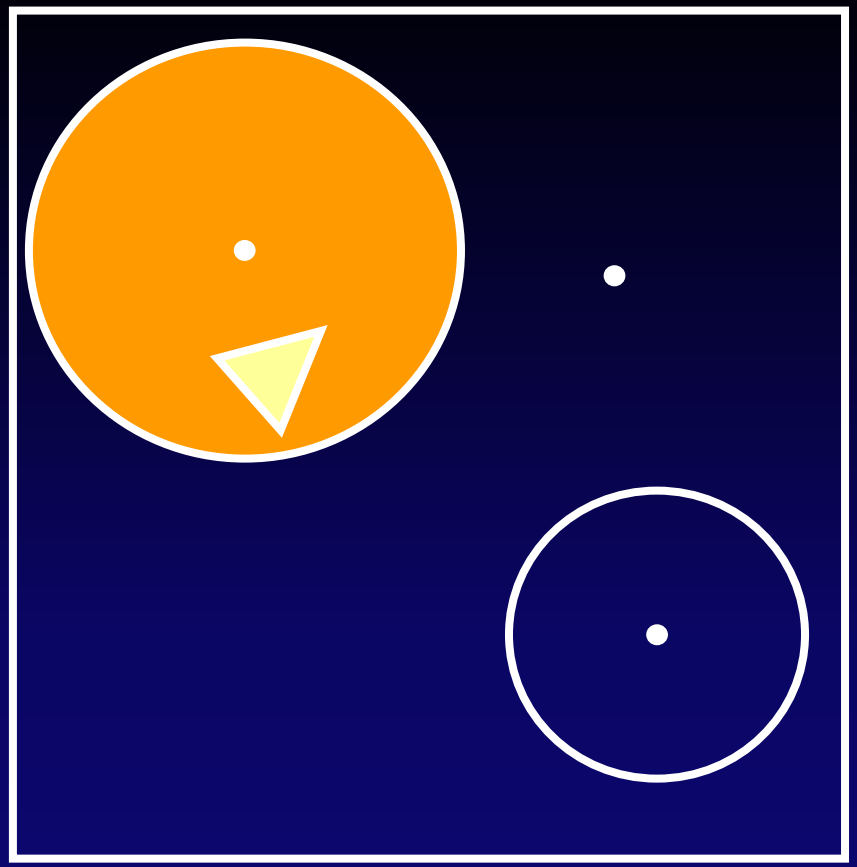
MIM: AOIM types

- Spatial
- Locales



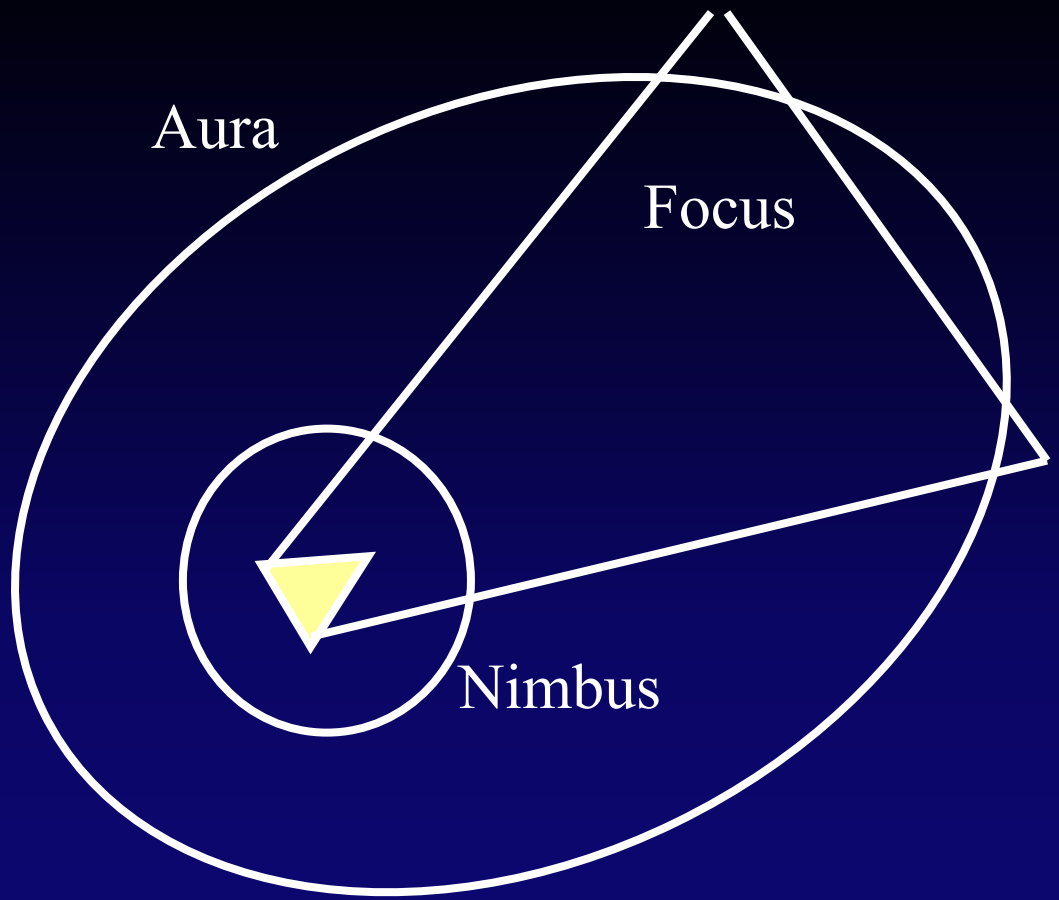
MIM: AOIM types

- Spatial
- Locales



MIM: AOIM types

- Spatial
- Locales
- **Auras**

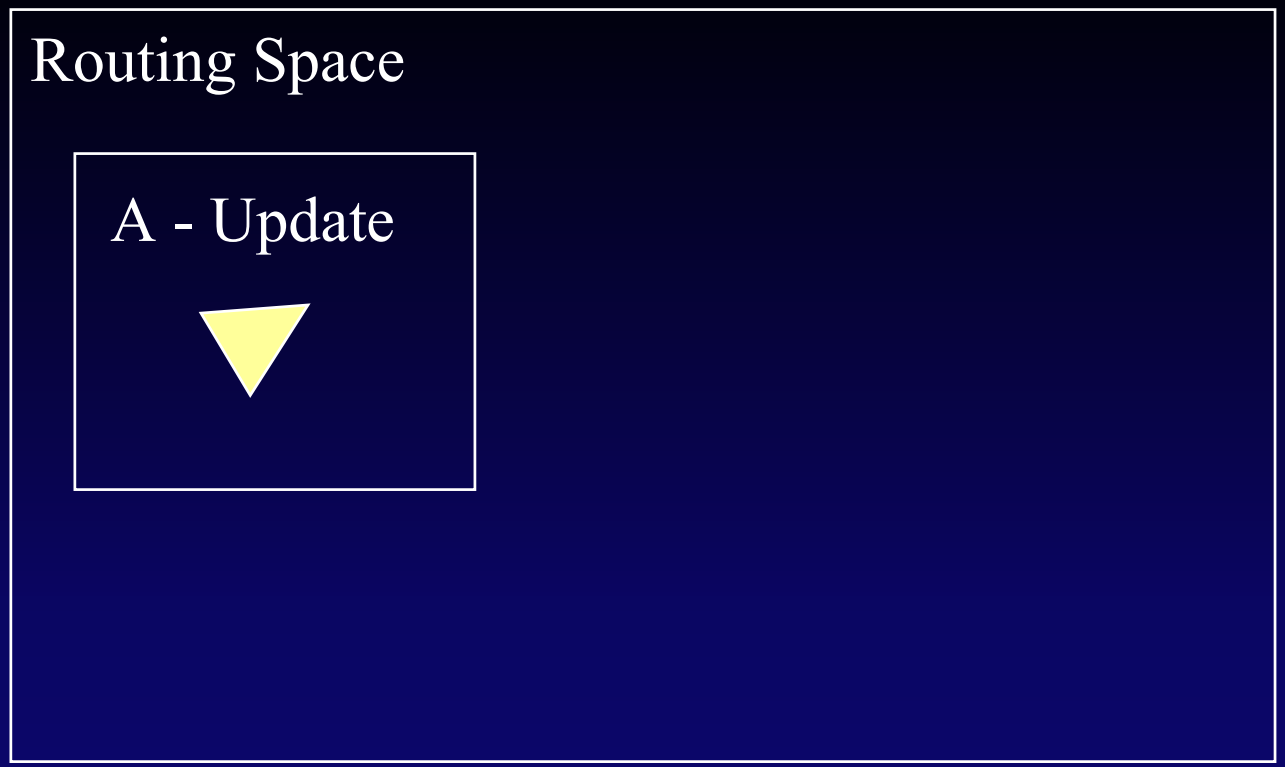


MIM: AOIM types

- Spatial
- Auras
- Locales
- Region

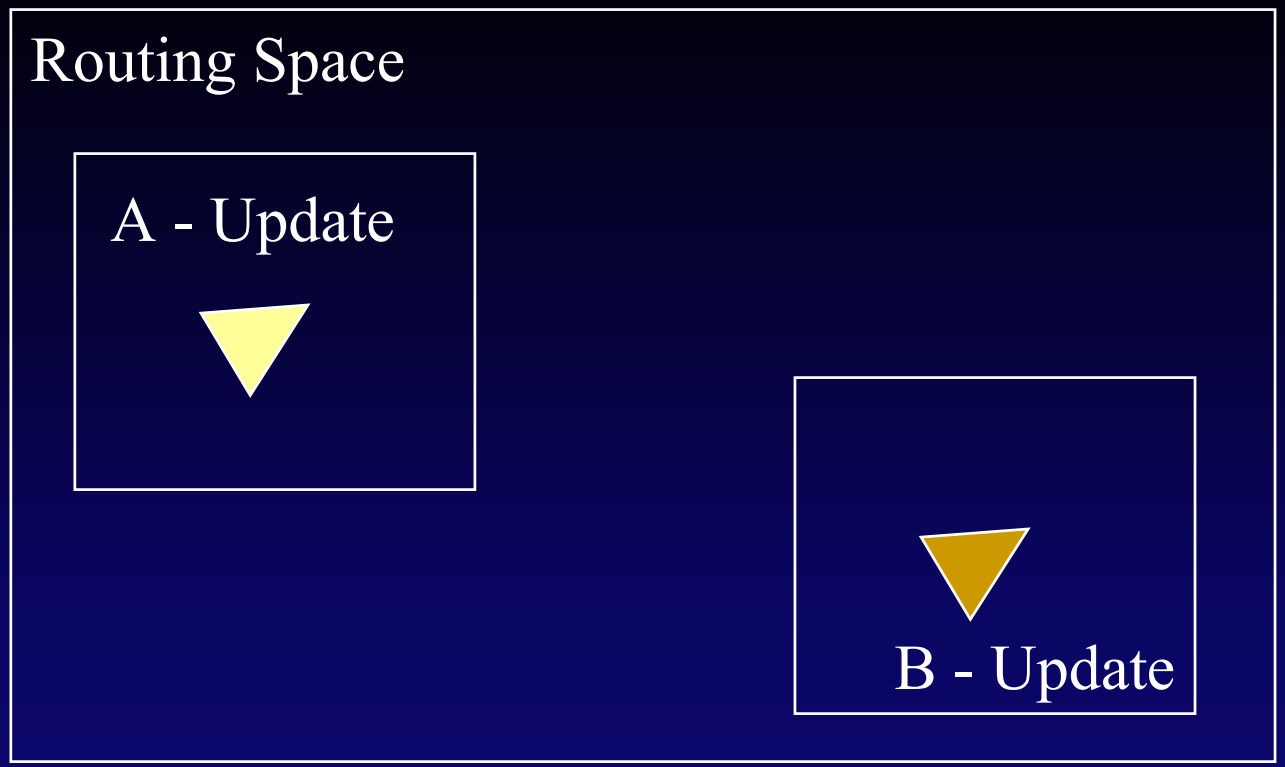
MIM: AOIM types

- Spatial
- Auras
- Locales
- Region



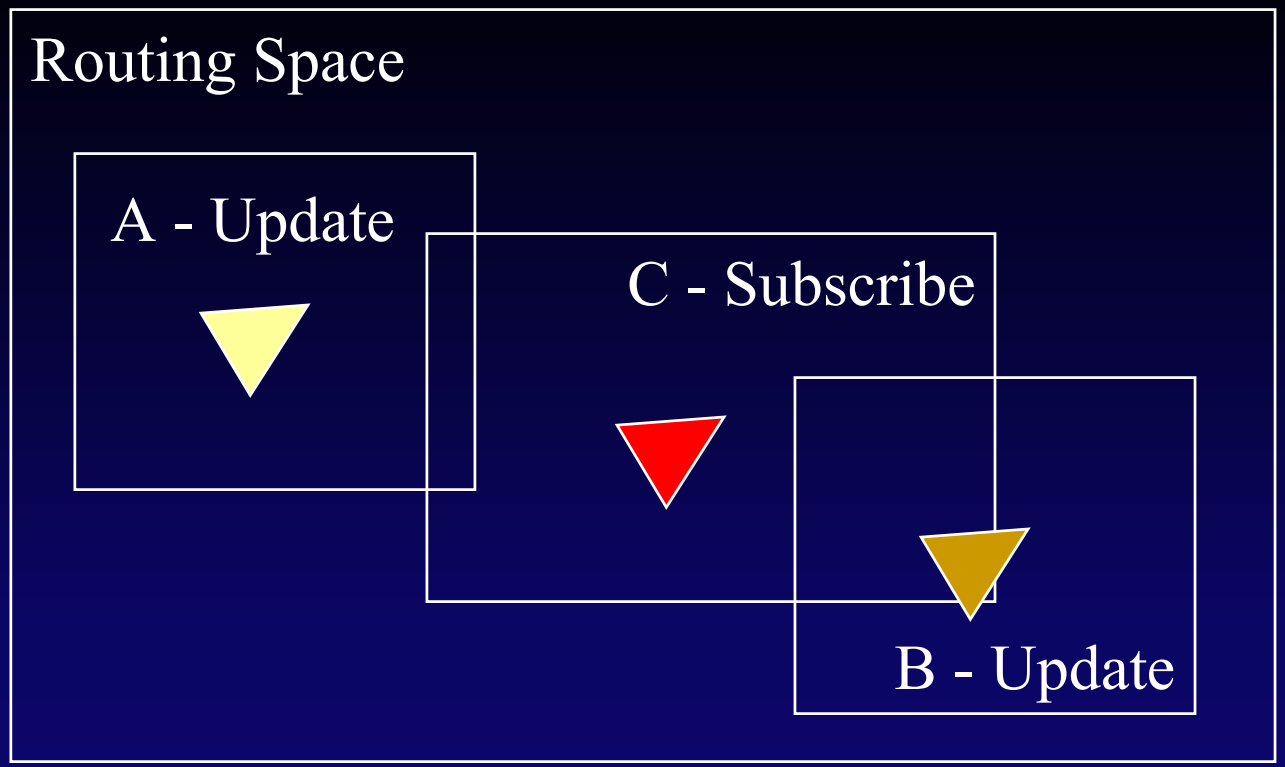
MIM: AOIM types

- Spatial
- Auras
- Locales
- Region



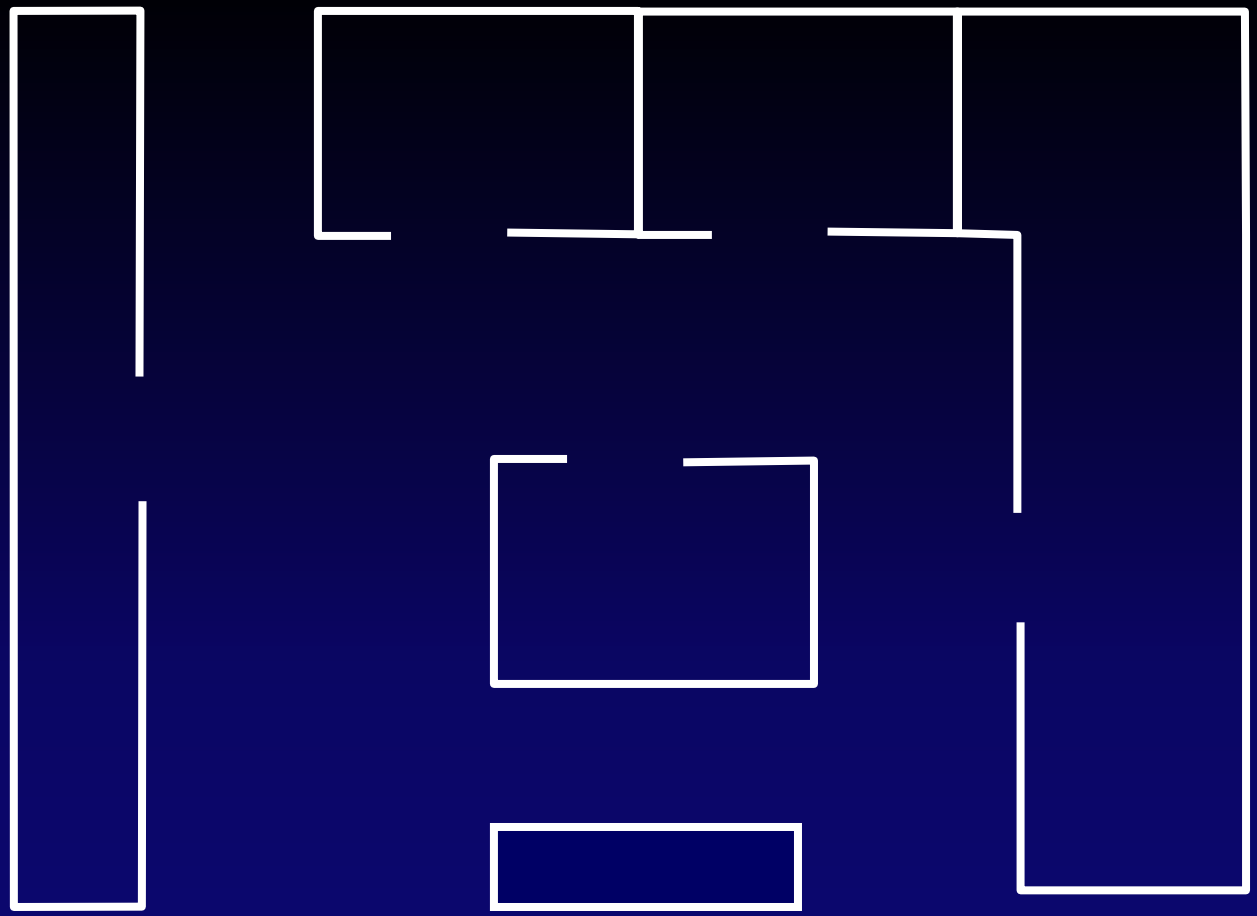
MIM: AOIM types

- Spatial
- Auras
- Locales
- Region



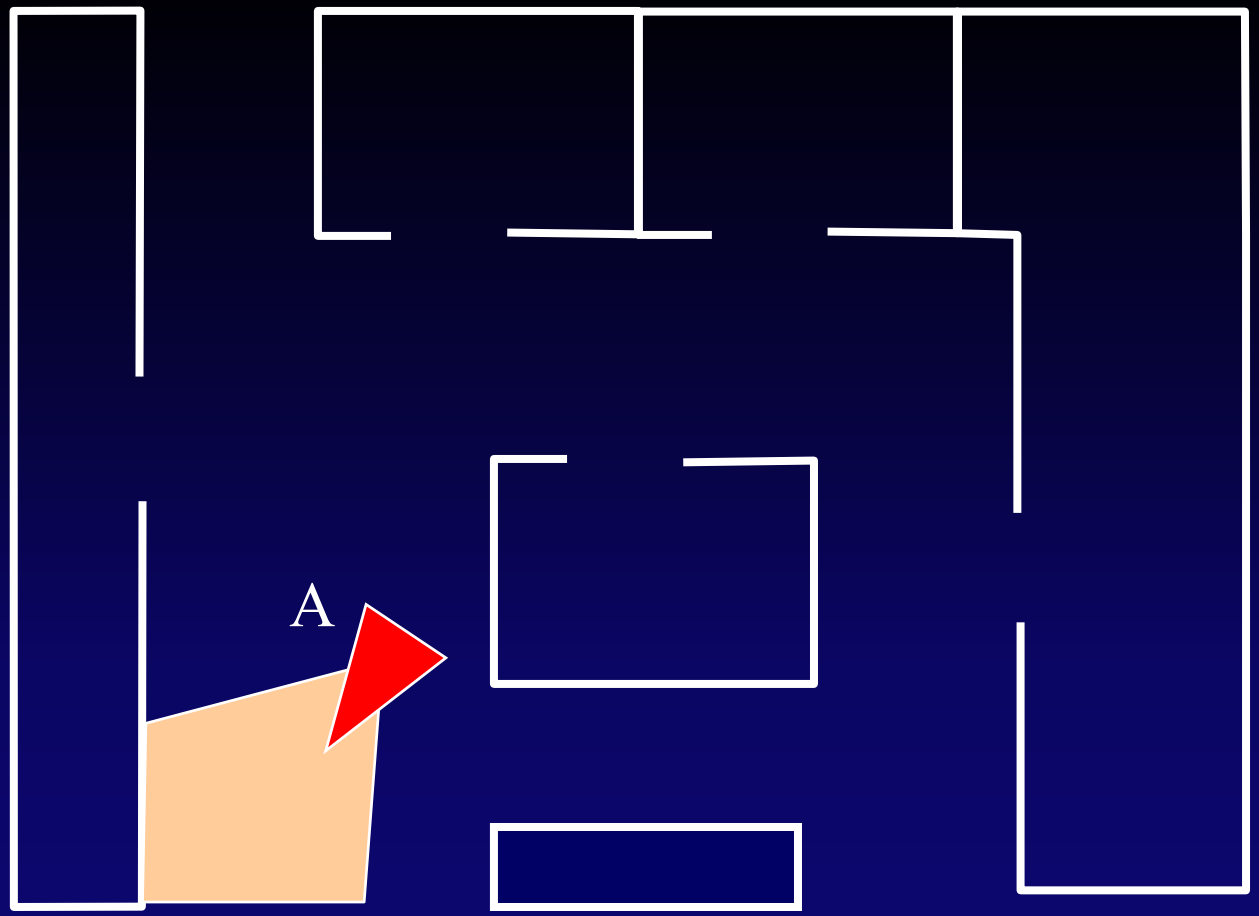
MIM: AOIM types

- Spatial
- Auras
- Locales
- Region
- Filtering



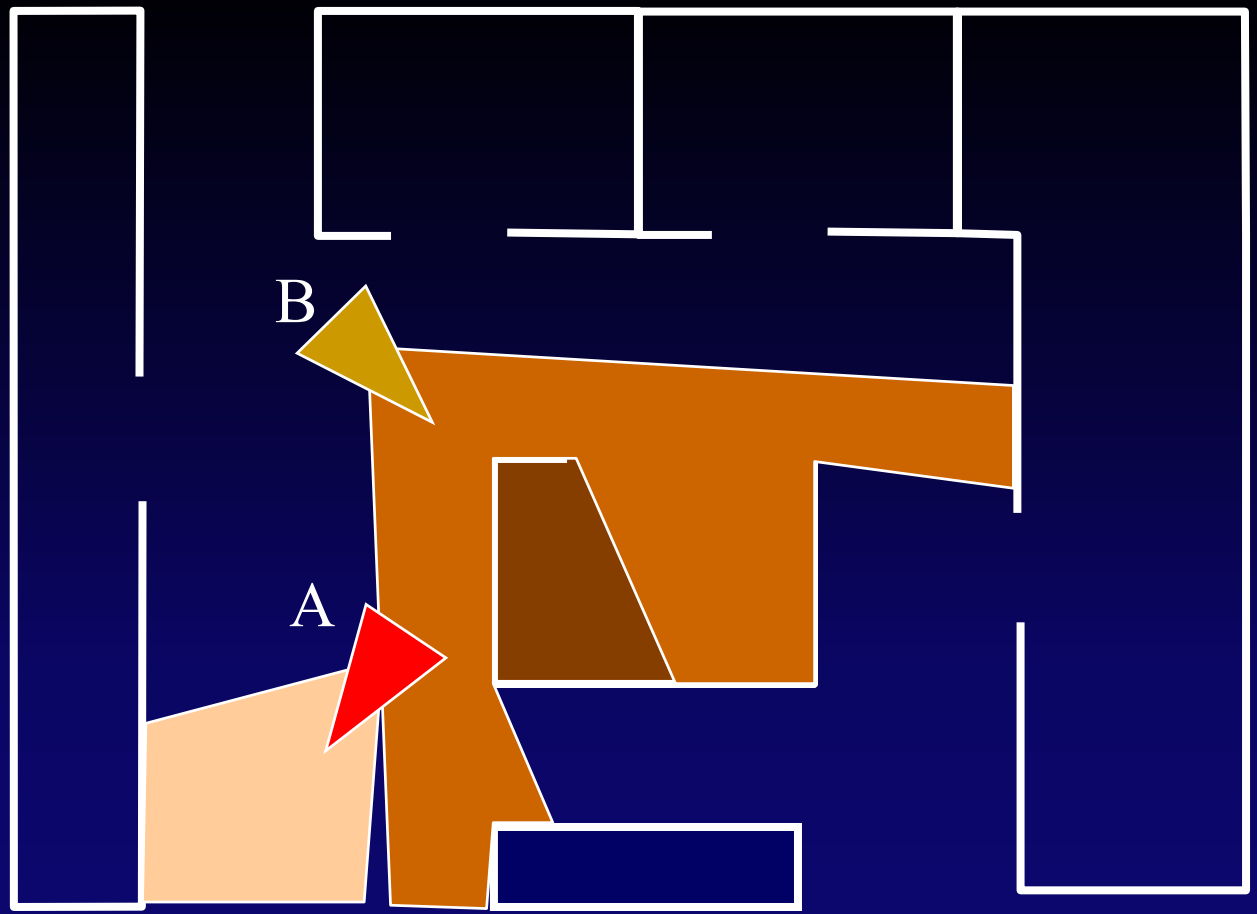
MIM: AOIM types

- Spatial
- Auras
- Locales
- Region
- **Filtering**



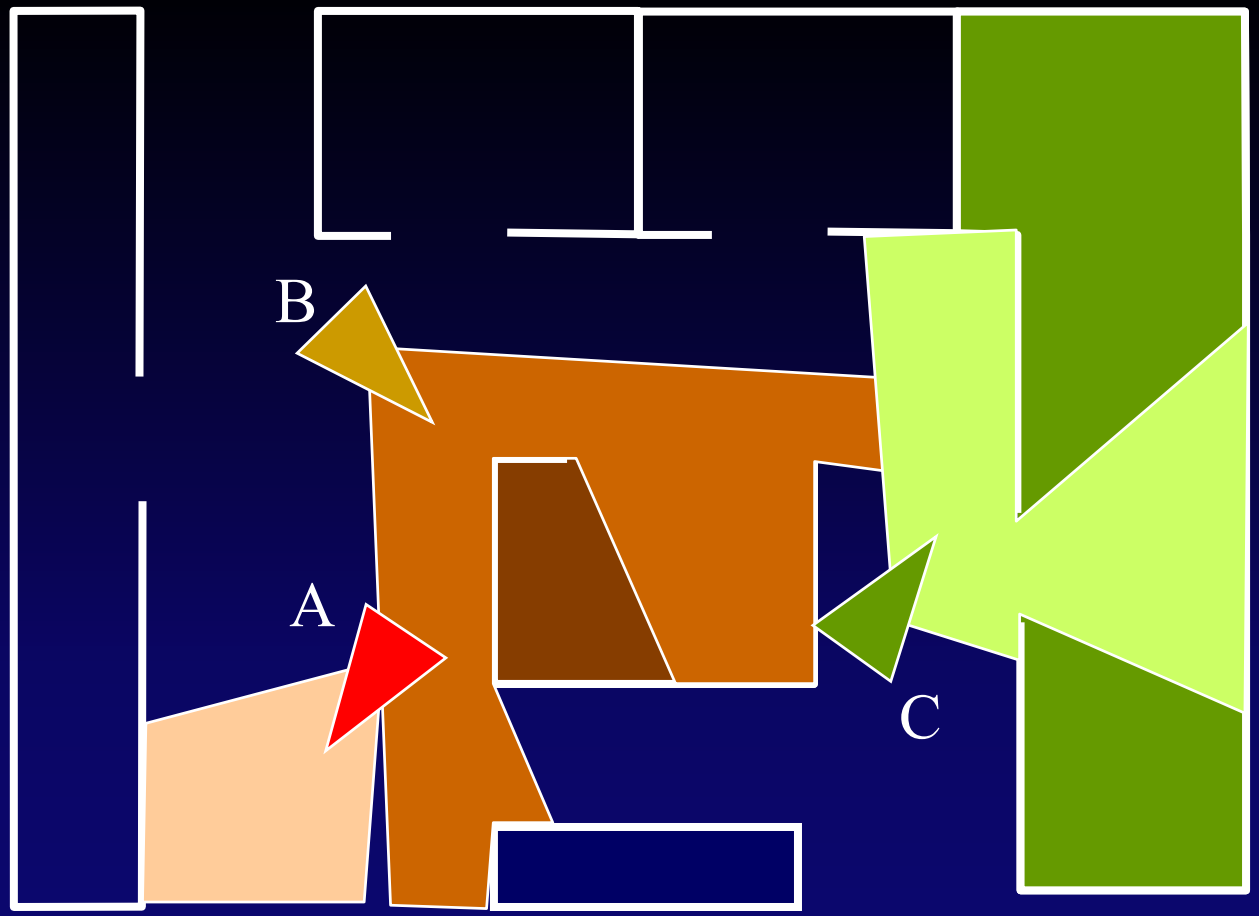
MIM: AOIM types

- Spatial
- Auras
- Locales
- Region
- **Filtering**



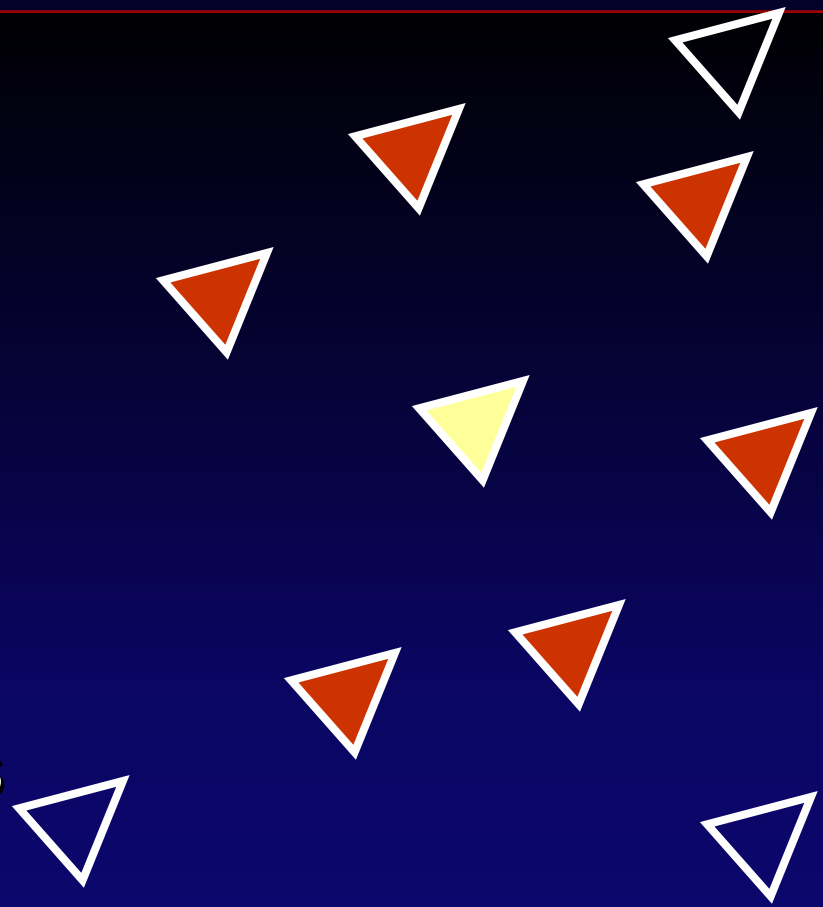
MIM: AOIM types

- Spatial
- Auras
- Locales
- Region
- **Filtering**



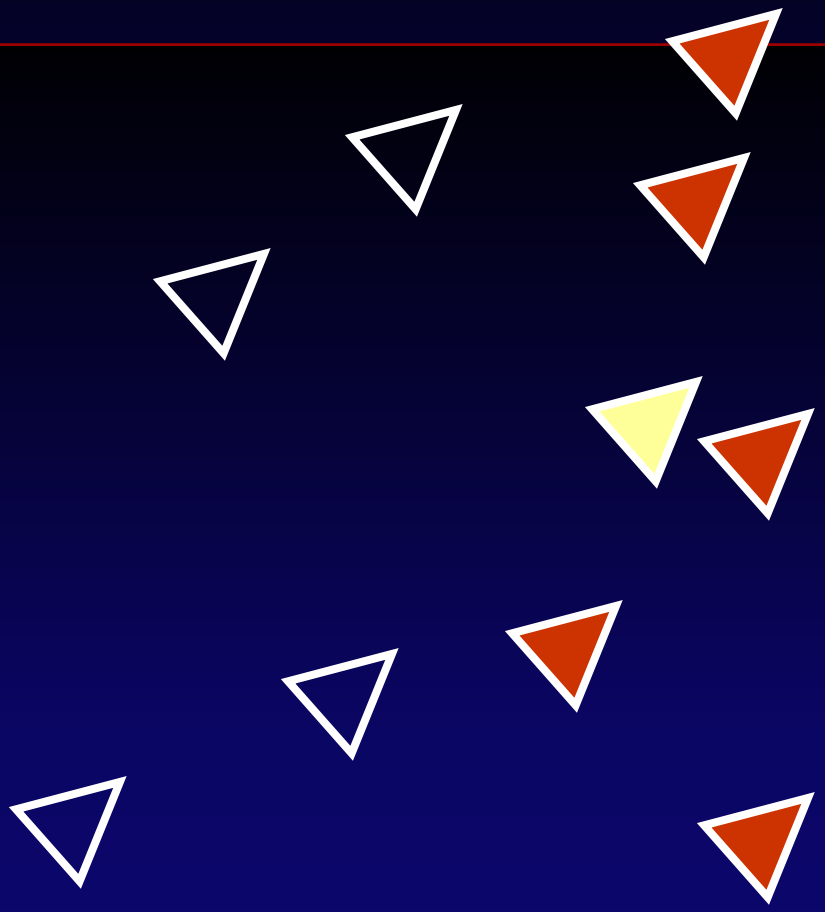
MIM: AOIM types

- Spatial
- Auras
- Locales
- Region
- Filtering
- **Nearest neighbors**



MIM: AOIM types

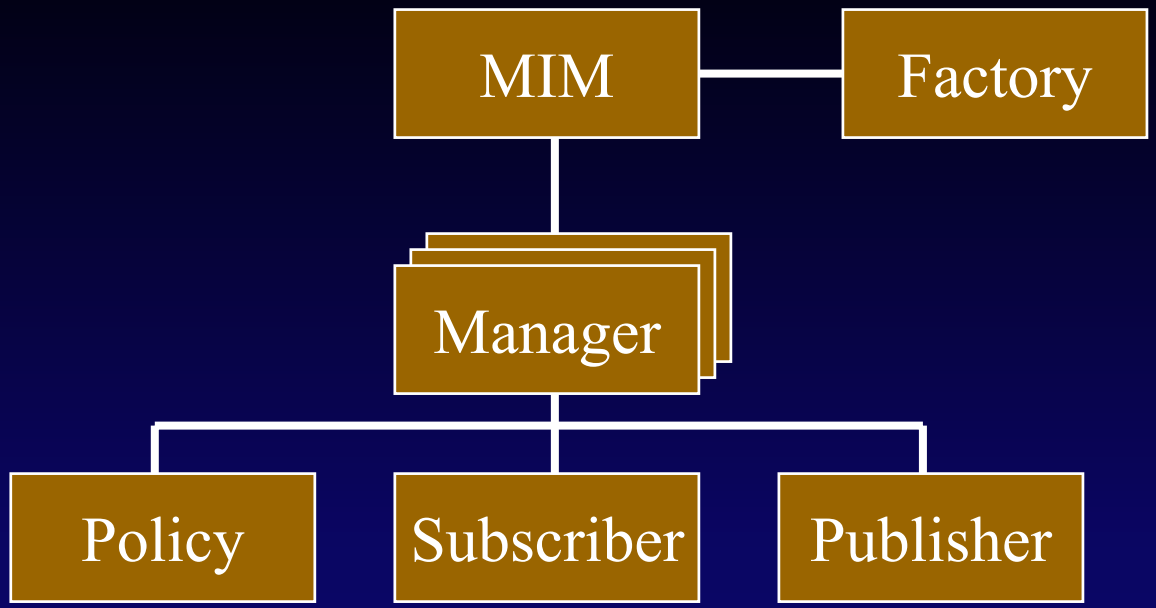
- Spatial
- Auras
- Locales
- Region
- Filtering
- **Nearest neighbors**



MIM: Common ground

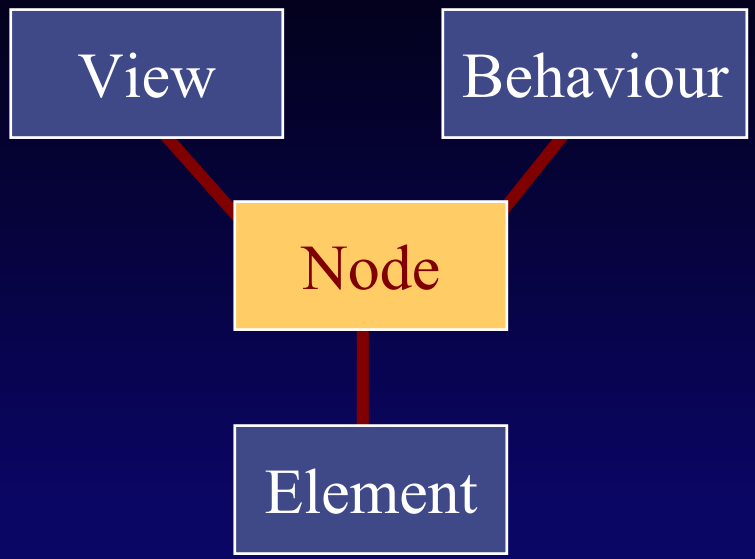
- Interest – scope information
- Wide range of policies
- Static vs Dynamic
- Client/Server vs distributed
- Poll vs Event based

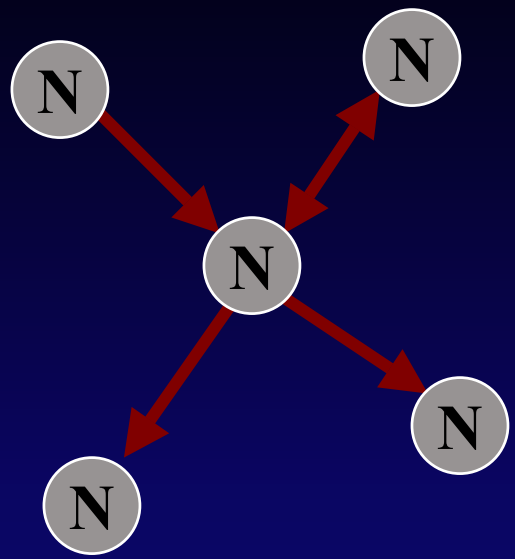
MIM: Core components



MUD: Meta Unified Datamodel

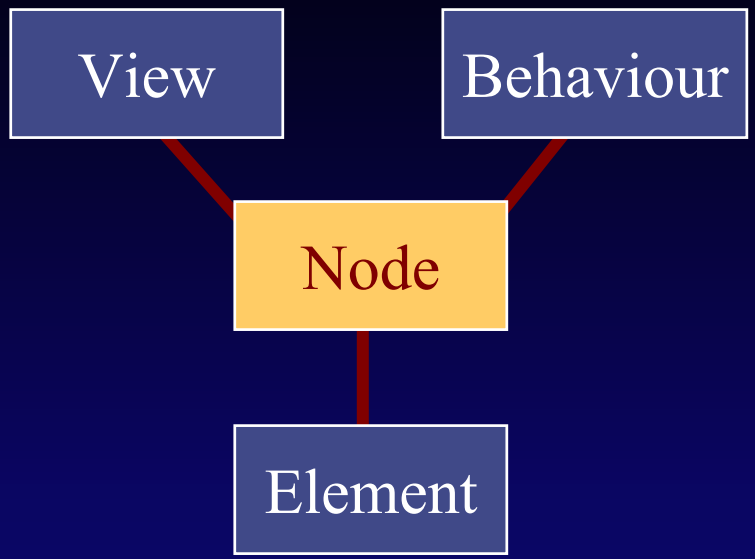
- Model-View-Controller pattern and Graphs

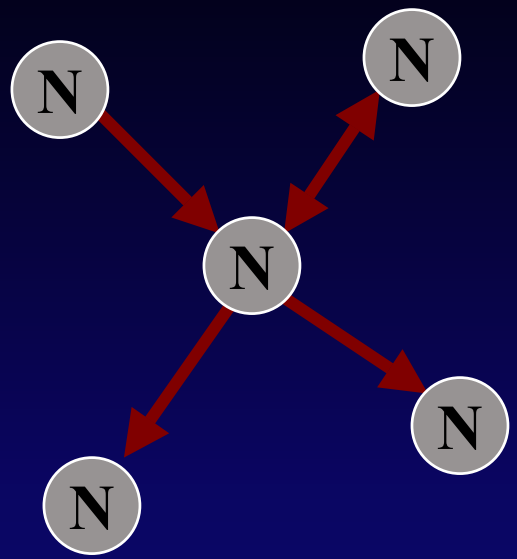




MUD: Meta Unified Datamodel

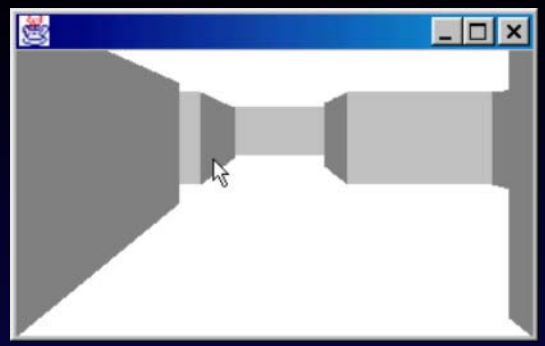
- Model-View-Controller pattern and Graphs



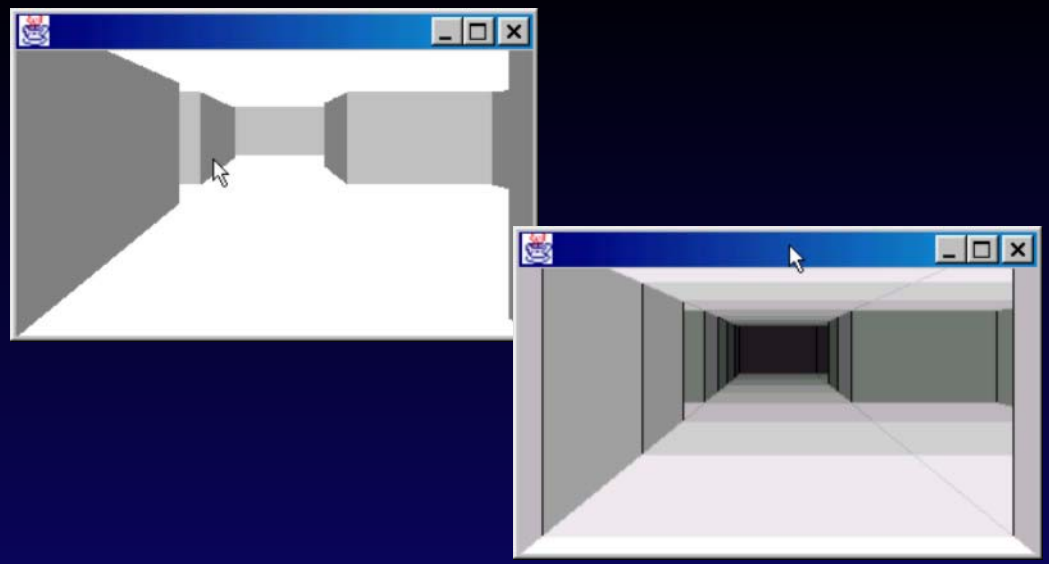


Client hetereogeneity

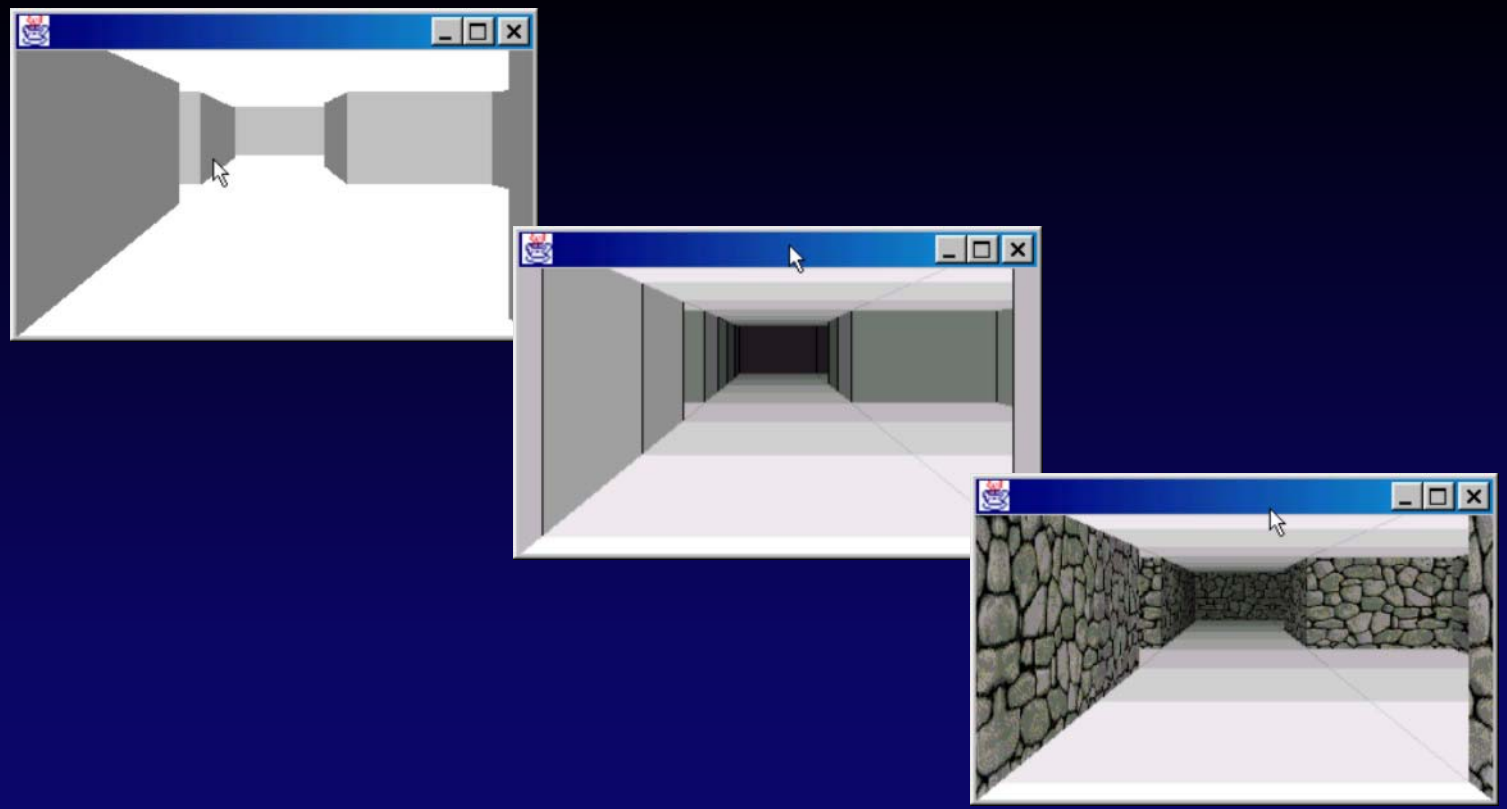
MUD: Meta Unified Datamodel



MUD: Meta Unified Datamodel



MUD: Meta Unified Datamodel



Q&A