- Material and some slide content from:
- Emerson Murphy-Hill
- Software Architecture: Foundations, Theory, and Practice
- Essential Software Architecture

#### **Architectural Styles** Reid Holmes

### MHI Pitch



### Prototype feedback



#### Deliverable #4 discussion



#### Blackboard vs Event-Based



#### Blackboard vs Event-Based



#### Peer-to-peer



# Style: Peer-to-peer

- State and behaviour are distributed among peers that can act as clients or servers.
- Components:
  - Peers (aka independent components).
- Connections:
  - Network protocols.
- Data elements:
  - Network messages.
- Topology:
  - Network. Can vary arbitrarily and dynamically.

# Style: Peer-to-peer

- Qualities:
  - Decentralized computing. Robust to node failures. Scalable.
- Typical uses:
  - When informations and operations are distributed.
- Cautions:
  - Security. Time criticality.



#### Architectural representations

- Characteristics of representations:
  - Ambiguity: Open to more than one interpretation?
  - Accuracy: Correct within tolerances
  - Precision: Exact but not necessarily correct
- Architectural models can be overwhelming
  - Model individual concerns with unique views
  - Views overlap and can be consistent on inconsistent



### Statechart diagram

- More formal description of system behaviour.
- Poor mapping between states and components.



# Component diagram

- Captures components and relationships.
  - Required and provided APIs explicitly recorded.



# Deployment diagram

Provide mapping between physical devices



## Sequence diagram

- Focus on inter-component collaboration.
- Capture behaviour for specific scenarios.



## Activity

- List the major components and connectors for your system
- Create a basic statechart for your system
- Create a rough component and deployment diagram for two different architectures
- Choose the arch that you think is best for your task; you will motivate this decision to me
- Perform an inter-group architectural review; be critical

