Architecting for COTS

- Component based systems
- Major architectural decisions can be independent of the components
- Use wrappers to augment off the shelf software to accommodate component architectures
Systems must be

- Predictable
- Able to evolve
- Scaleable
- Recoverable
- Administerable
System as Network

- Some components wrap COTS
- Others are custom
Predictable

- Model for concurrency
- Synchronization
- Component isolation / protection
- Minimize regression testing
Able to Evolve

- Do not disturb interfaces that work!
- Wrapper isolates changes to COTS
- New capabilities appear as new services from the wrapper
- Only components which are users of new services need be changed.
- Hot plug-in of components
Scaleable

- Use multiple copies of same component
- Separate processes from resources
- Centralize resource allocation
- Reliability
- Hardware - software tradeoffs
Recoverable

- Component State
- Configuration (component relationships)
- Transactions (inter-component state)
- Monitoring
- Third party connection model
Administerable

- Instrumentation
- Monitoring
- Asynchronous events
  - Errors
  - Start, quiesce, shut down