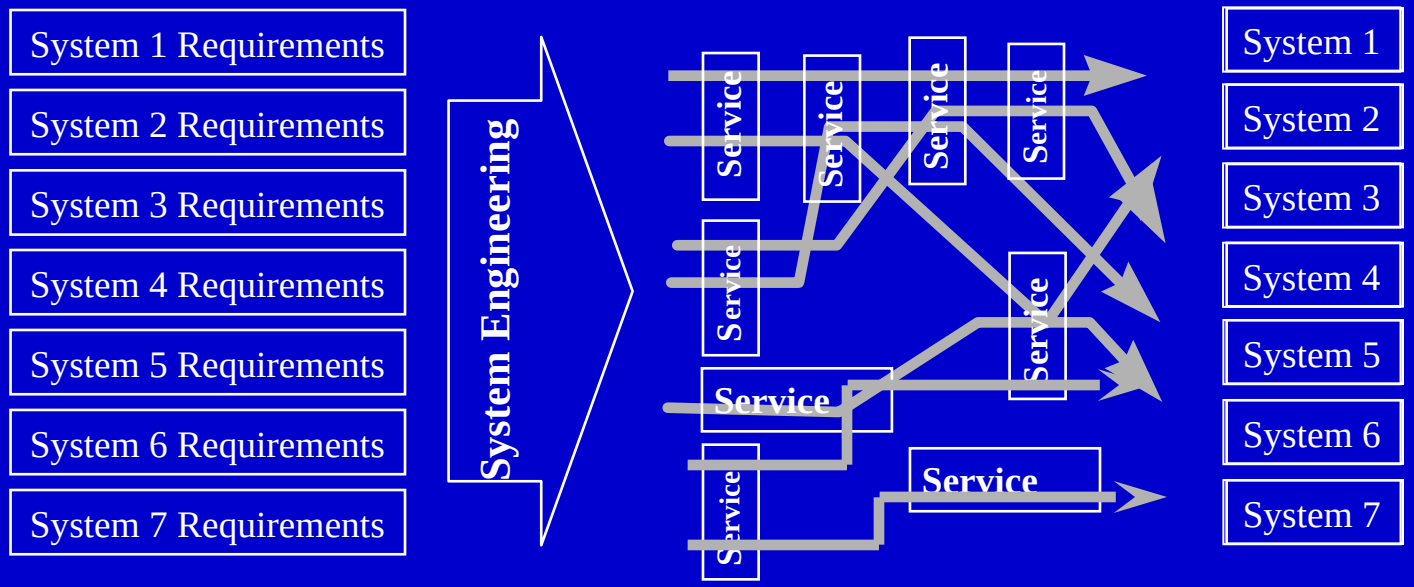
Unified Cryptologic Architecture: A Framework for a Service Based Architecture

Howard Winter
Department Of Defense
hkwinter@ncsc.mil

Agenda

- Service Based Architecture Description
- Benefits of a Service Based Architecture
- Cryptologic Framework Principles
- Cryptologic Framework Design
- Enterprise Scalability Schedule
- Participating Projects
- Framework Runtime Environment/COTS

Service Based Architecture



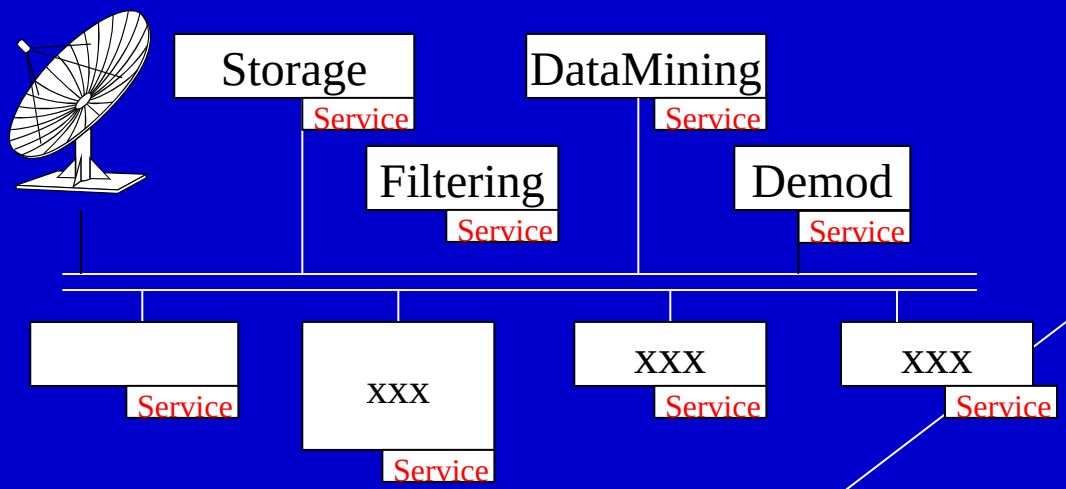
Benefits Of Service Based Architecture

- Interoperability
- System upgrades substantially easier
- Enables shared resources
- Developers Concentrate on Their Service
- Substantial Productivity Gains Based Upon Experiences

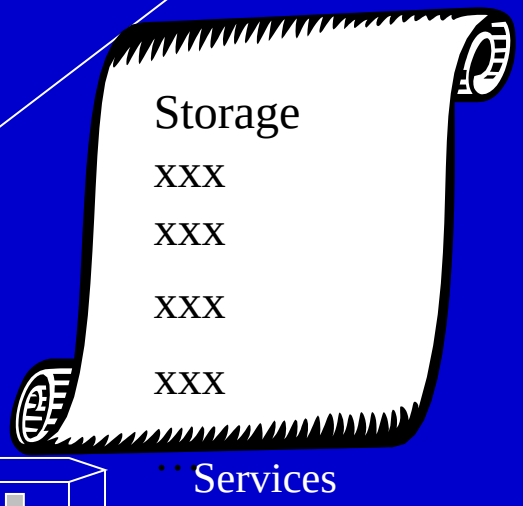
In Private Industry (AT&T, Motorola, Ericsson, HP, and others)

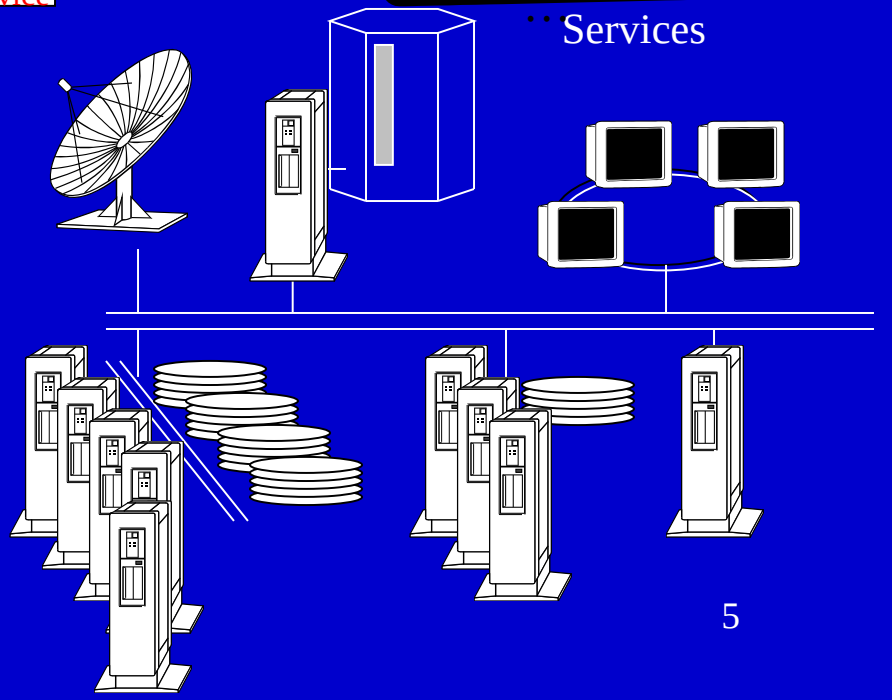
- Time To Market Reductions of 2 to 5 Times
- Defect Density Reductions of 5 to 10 Times
- Maintenance Cost Reductions of 5 to 10 Times
- Overall Software Development Cost Reductions of 15% to 75% for long term projects
  - Includes the Overhead Cost of Developing Reusable Assets and Supporting Their Use

System Integration



Local System





Shared Resources

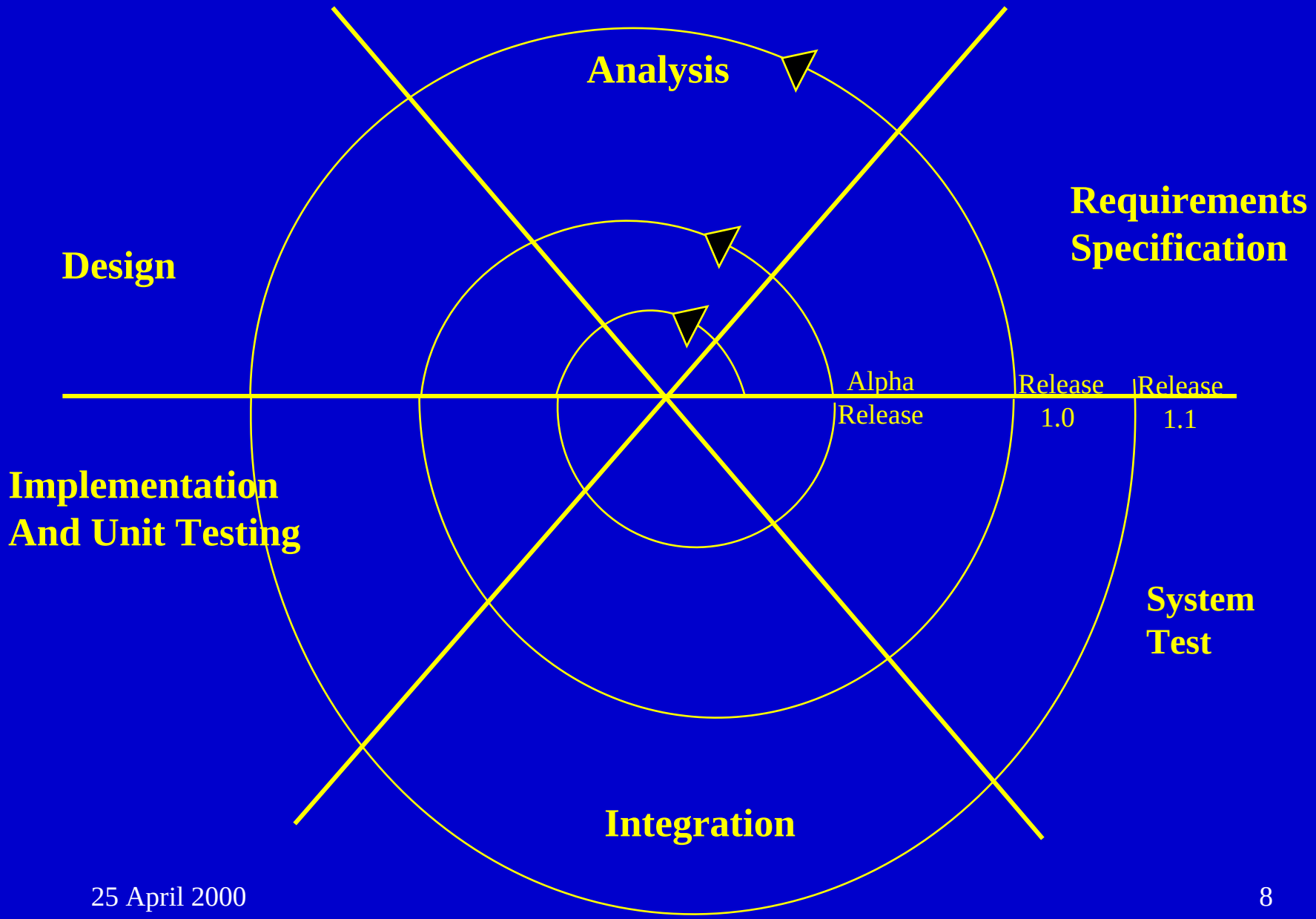
Object Oriented Frameworks

- A set of cooperating classes that make up a reusable design for a specific problem area
- Becoming increasingly common and important; way OO systems achieve the most reuse
- Allows apps developers to concentrate on the specifics of their applications
- Dictates architecture of application; emphasizes design reuse over code reuse
- Usually includes concrete subclasses that can be put to work immediately
- You reuse the main body and write the code it calls
- Large object oriented developments consist of layered or integrated frameworks
- Hard to design; need to be applied to at least three applications

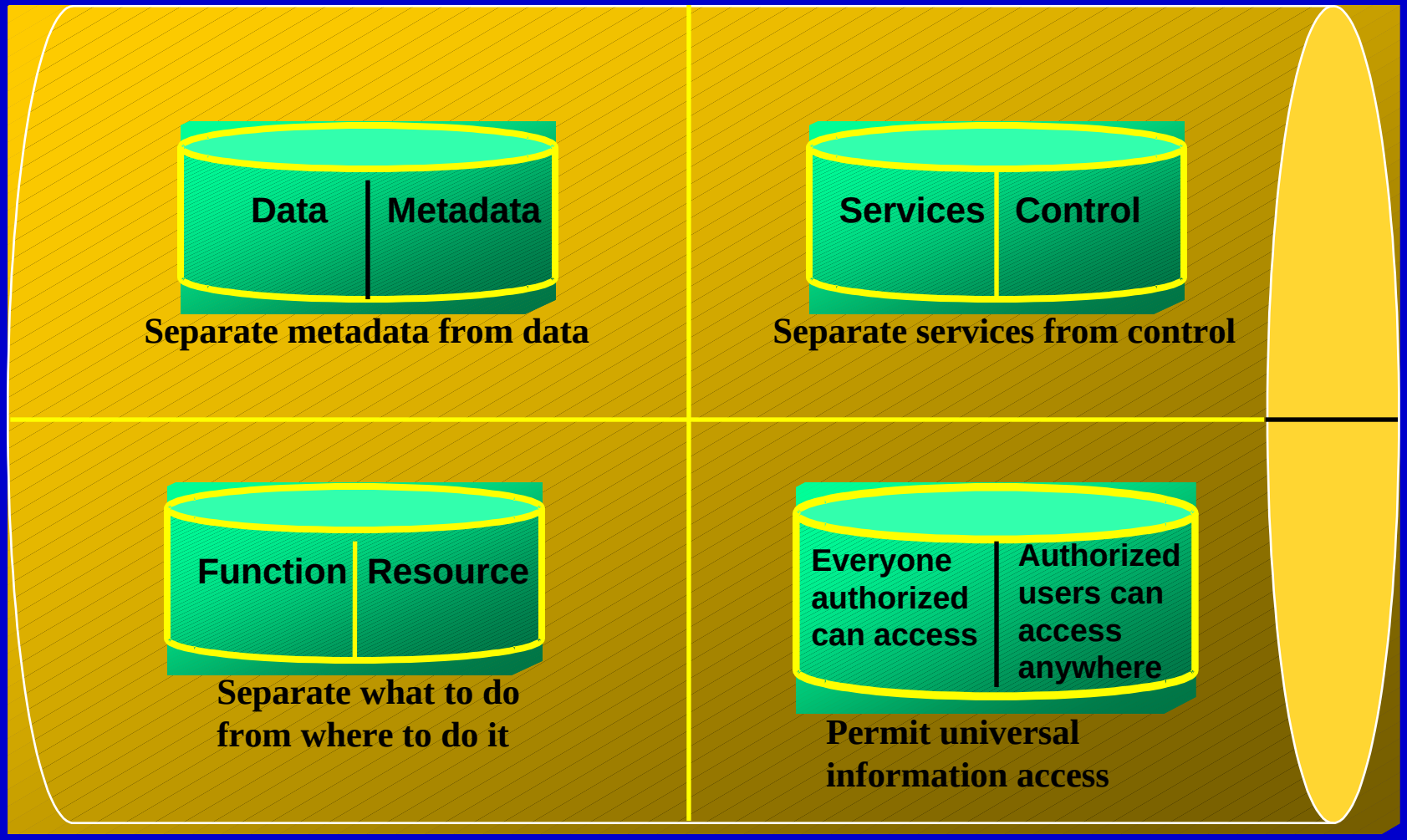
Cryptologic Framework Concepts

- Develop software system such that multiple Mission and Infrastructure Services can be integrated together to meet DO requirements.
- Will Lead To “Plug And Play” SIGINT Services
- Will allow system developers to field systems:
  - Cheaper
  - Quicker
  - With Fewer Defects
- Framework will be phased in with pilot and other applicable projects.
- Framework is key enabler for a Service Based Architecture
- Framework is developing many generic services that can be reused by many projects

Incremental Development Model



Cryptologic Framework
Underlying Principles



Service Based System Design Using the Framework

Services/ Registry

- S1 - R1
- S2 - R2, 3, 4
- S3 - R8
- S4 - R6, 7
- S5 - R9
- S6 - R2,3,4,5
- S7 - R2
-
-
-

Resources

- Hardware Demux (R1)
- Cray (R2)
- Dec Alpha (R3)
- Pentium III NT (R4)
- SGI (R5)
- Sun Ultra (R6)
- Sun Enterprise (R7)
- Hardware Demux2 (R8)
- Vax (R9)
-
-
-

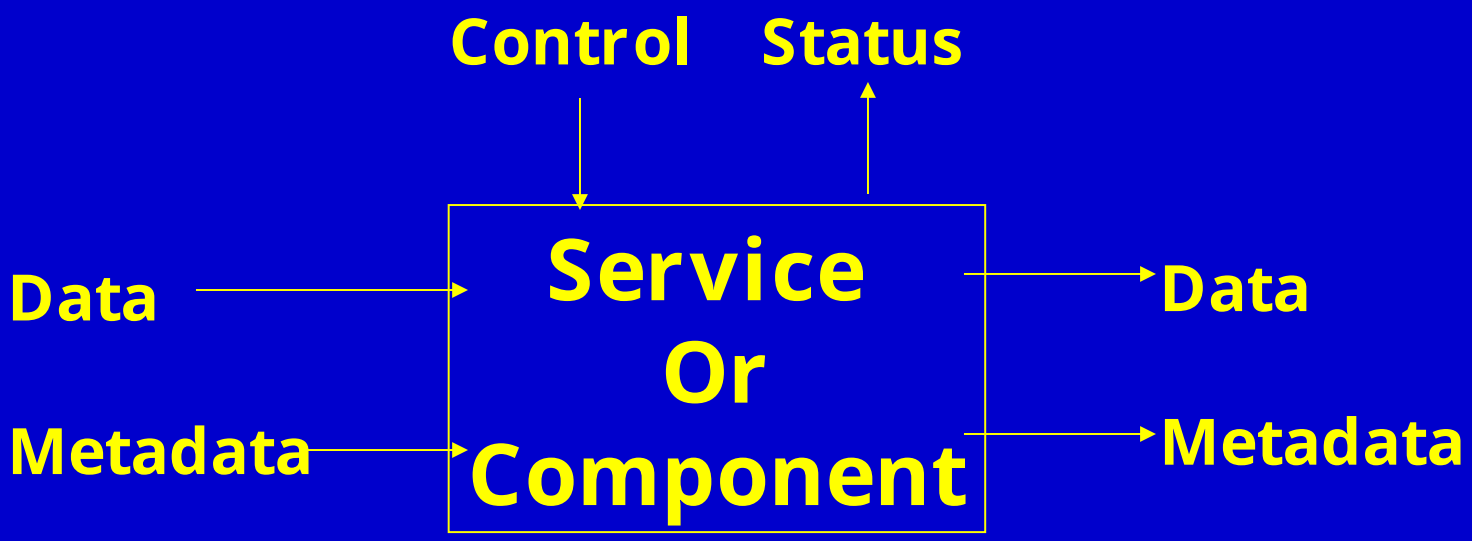
Data/Metadata

- Standard format for each service
- Available via access service and storage

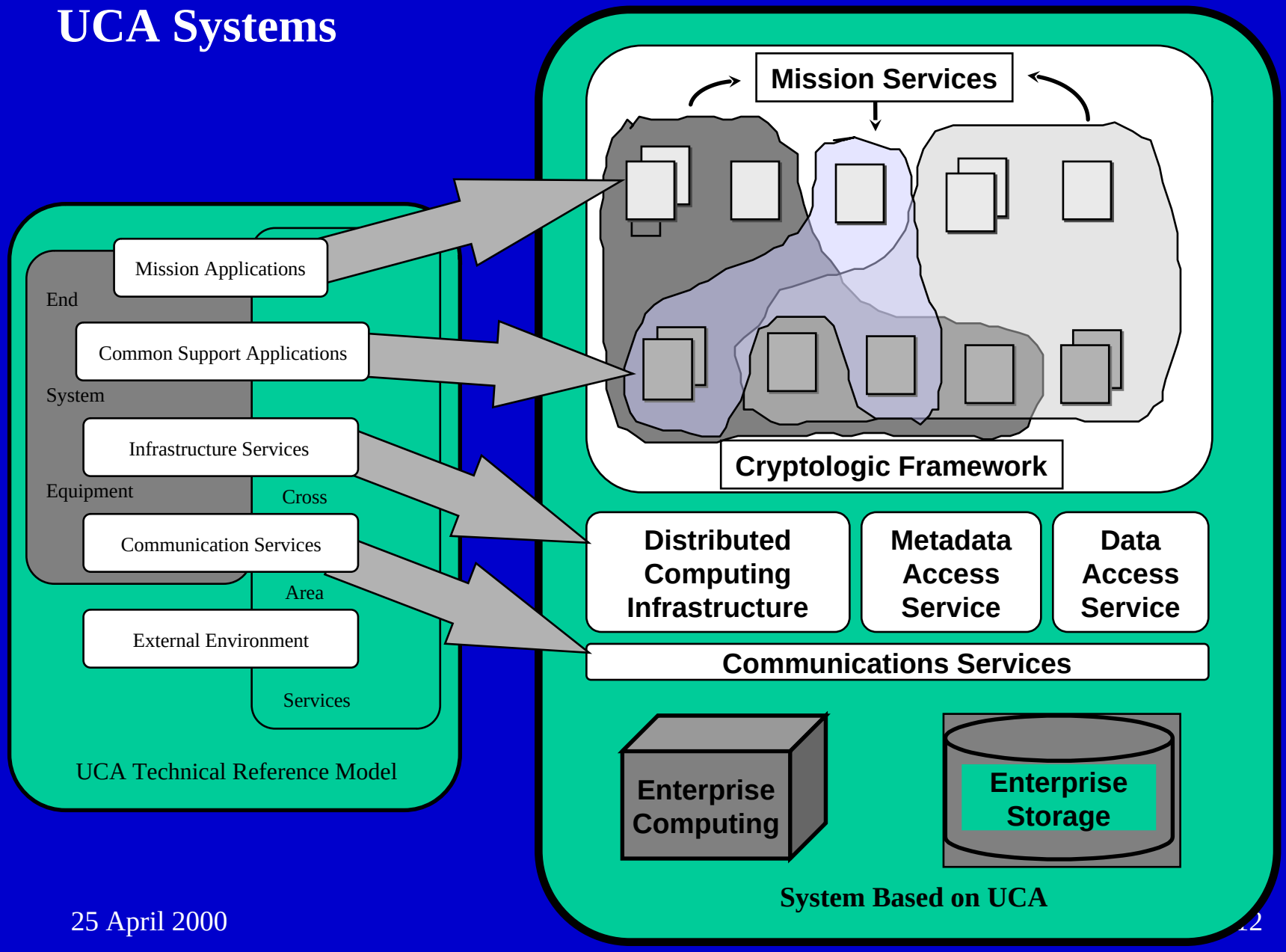
Business Rules (Control Logic)

- S4
- S2 on R2
- S7
- S2 on any available resource
- S1
- S6 on R2 or R3
-
-
-

Basic Service or Component



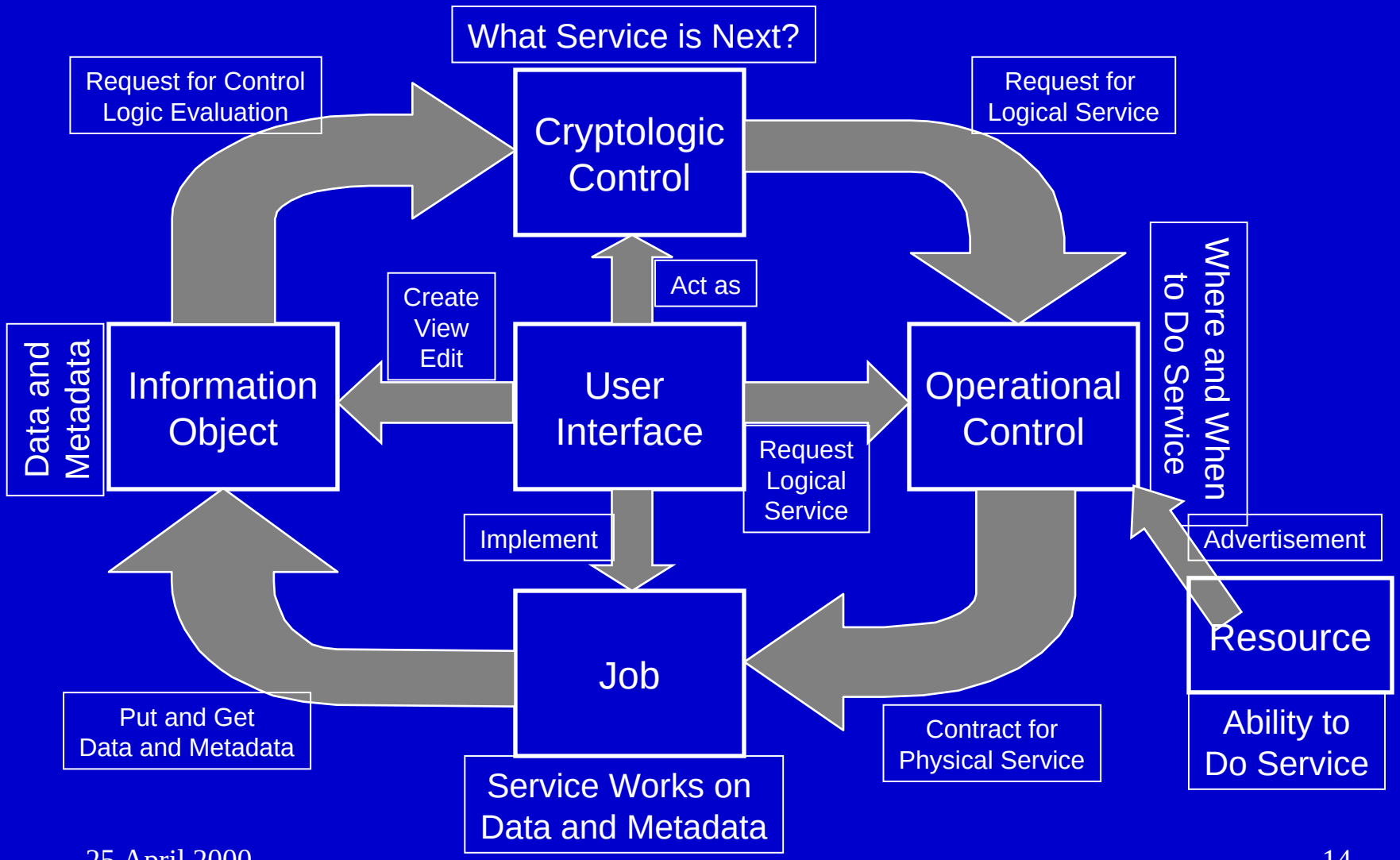
Implementation of UCA Systems



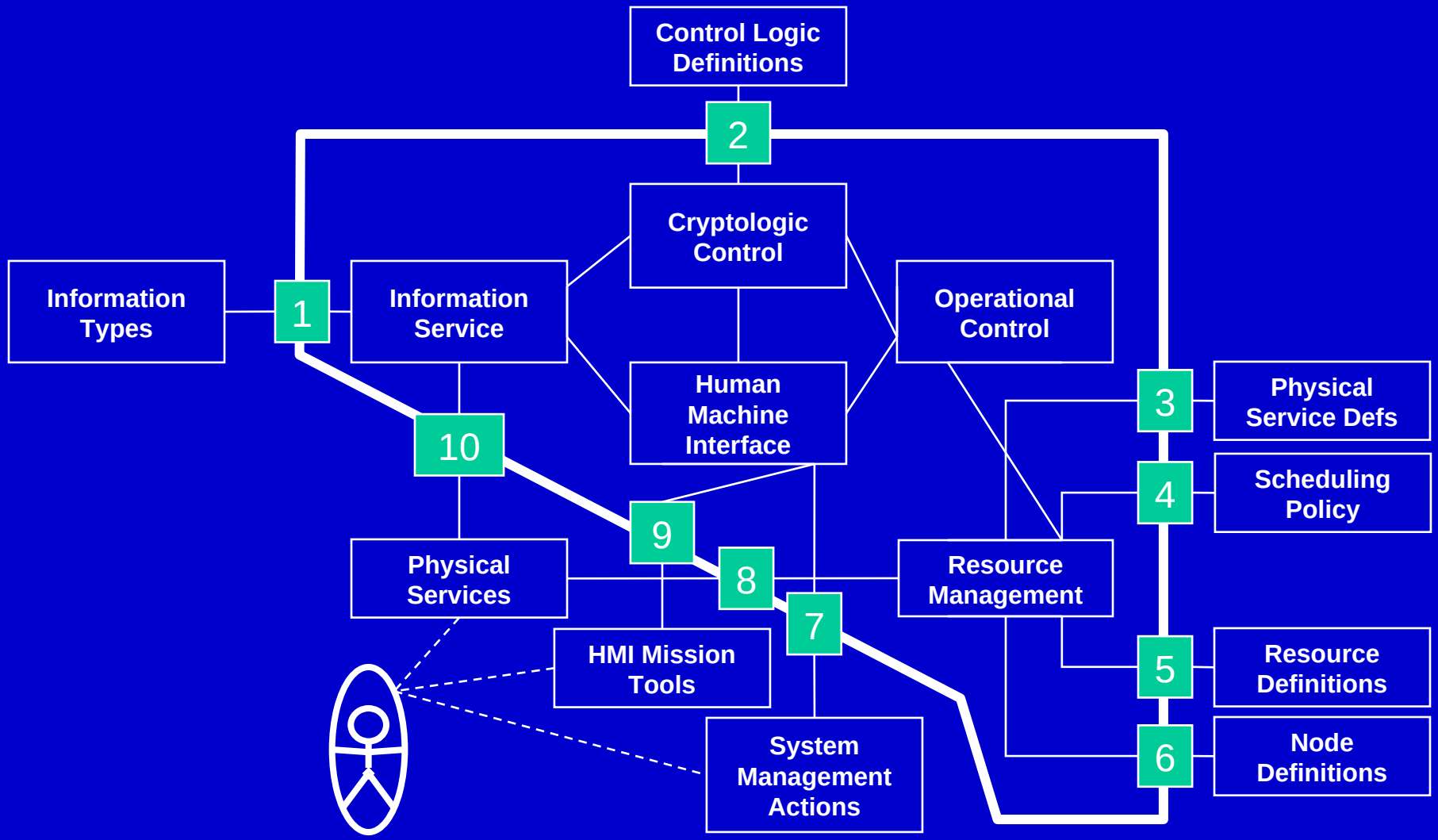
Three Tier/N - Tier Architecture

(Widely accepted in industry)

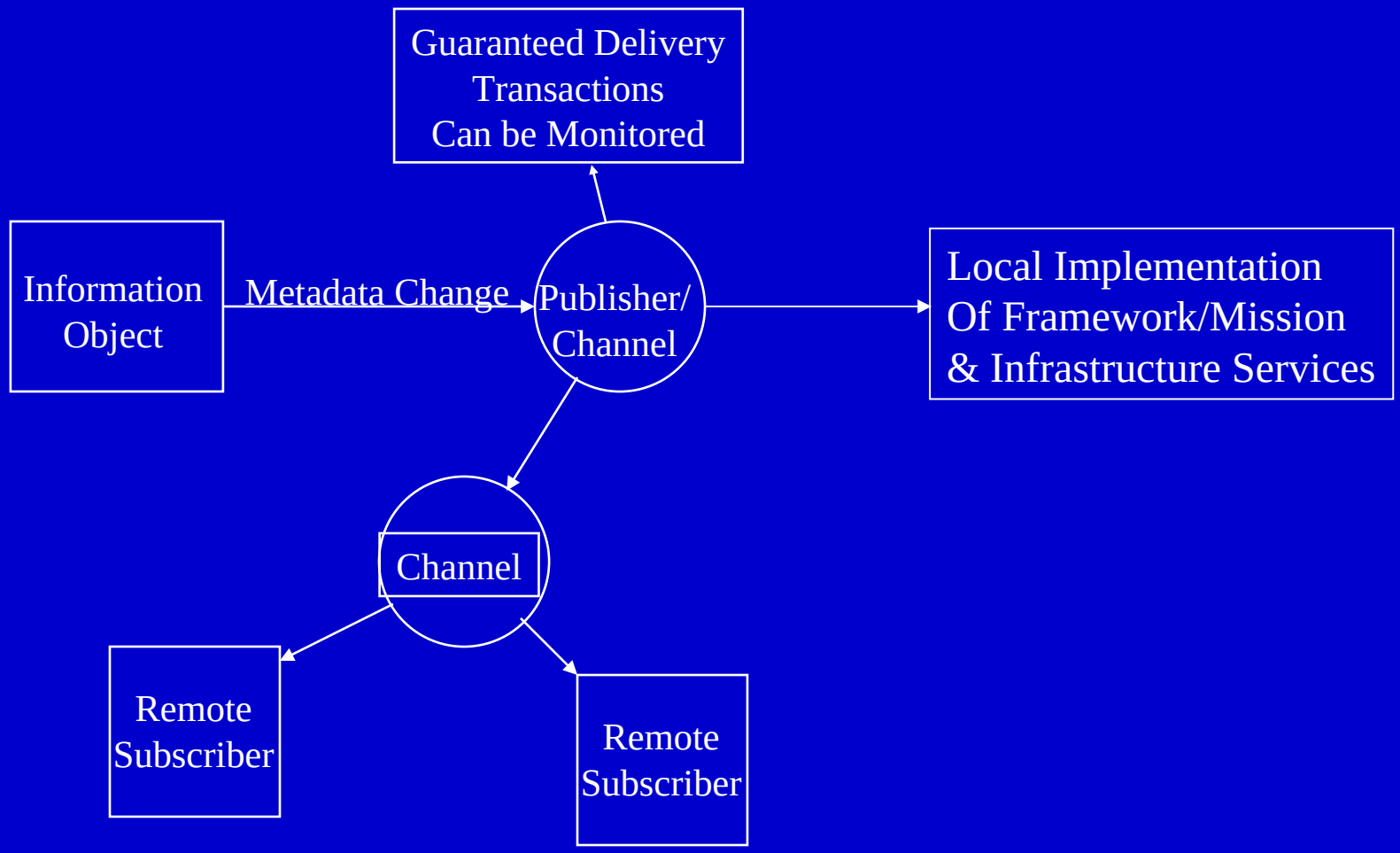
Framework Key Concepts



The Framework Inside and Out



Publish and Subscribe Messaging Service (Provides Scalability of Processing)



Enterprise Scalability Schedule

Release 1.0	Feb 00	Minimal scalability
Release 2.0	June 01	Initial Federation of Projects
Release 3.0	July 02	Site Support
Release 4.0	July 03	Enterprise Support
Release 5.0	July 04	Partner Support

Note: Release 2.0 and beyond are estimated dates.

Ways The Framework Will Be Used

- Use the entire integrated implementation of the Framework
  - Integrate physical services and information objects into mission services
  - Integrate mission services together to develop delivered systems
- Use some of the Framework Applications and extend or reimplement some on your own in order to satisfy specific requirements such as performance or capacity.
  - Follow the interfaces of the UCA Framework
  - Then integrate services together as above
- Implement the interfaces of the UCA Framework with different mechanisms.

Building Systems Using The Framework

- New systems consisting of mission and infrastructure services are integrated using the framework
  - Draw on rich set of re-usable corporate services
    - Extend existing services or develop new services if necessary
  - Define Control Logic
  - Define Data/Metadata
  - Size and purchase hardware
  - Plug services into Framework

Participating Projects
(Using the Framework)

- **Seven Committed Projects**
- **Many Other Potential Projects**
- **10’s Of Mission Services Currently Being Developed**

Run-Time Environment

- **Beta**
  - **Whole Framework**
    - **Solaris 2.6**
  - **Services**
    - **Solaris 2.6 (C++, Java)**
- **Release 1.0**
  - **Whole Framework**
    - **Solaris 2.6/7?, NT?**
  - **Services**
    - **Solaris 2.6/7?, Digital Unix, Windows NT, SGI, HPUX**
    - **Others as needed**

Framework Runtime Commercial Products

- Analysis and Design in UML
- Language: Java
- Software Metrics
- CORBA
- Messaging Service
- Enterprise Java Beans (EJB) Application Server (Release 2)
  - Security Products
- Resource Management (Release 2)
- Database