PanSTARRS Data Segment

Concept Design Review

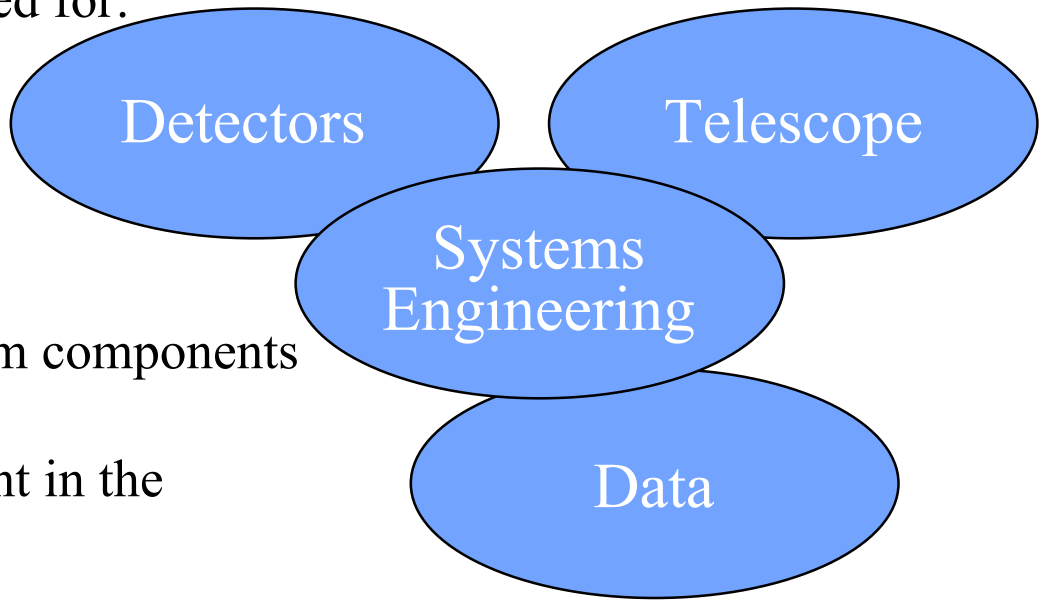
February 2003

Topics

- What is the data segment?
- Current concept of data flow
- Data processing
- Data management

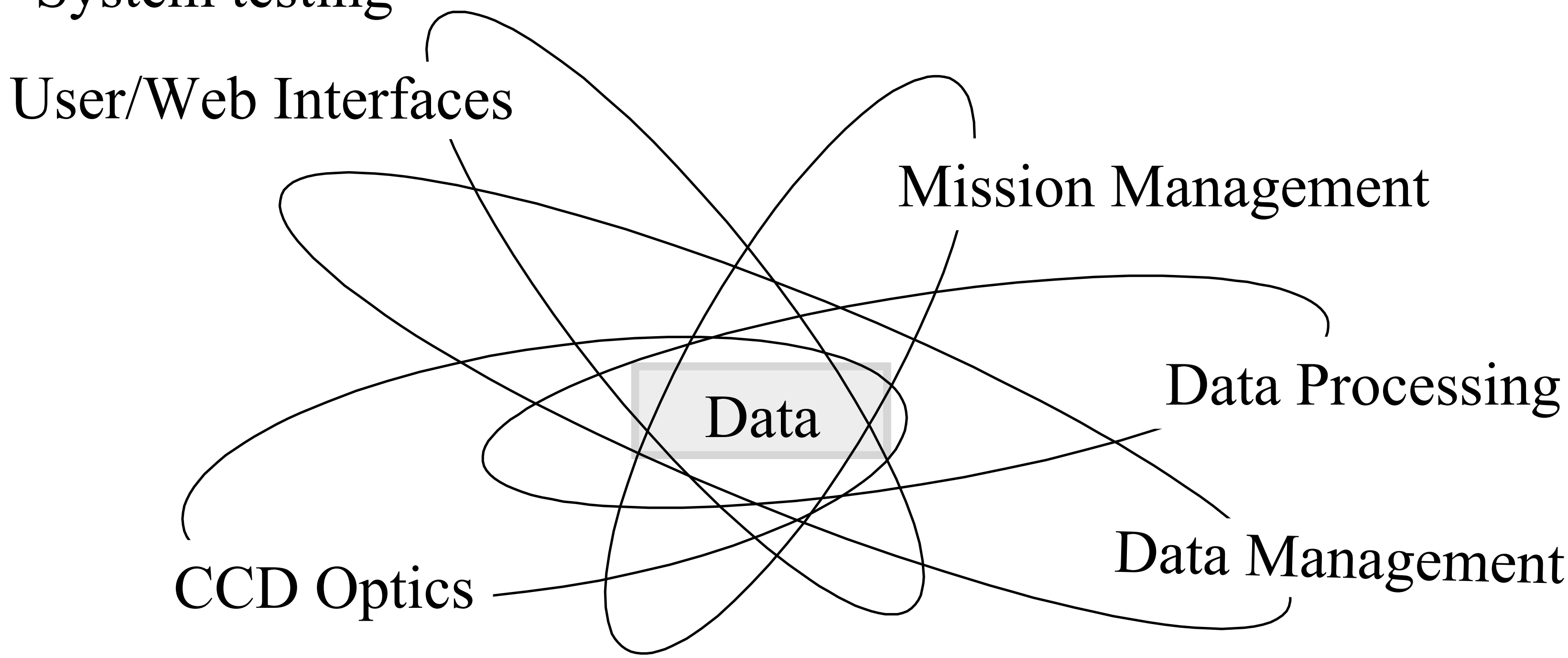
What is the Data Segment?

- Components and interfaces required for:
  - Data analysis
  - Data modeling
  - Data processing
  - Data storage
  - Data retrieval for other system components and user interface
- Data is integral to every component in the system
  - Telescope acquires data
  - Detectors output data
  - Data processing manipulates data
  - Data storage retains and accesses data for other components and user interface
- Therefore, systems engineering is intimately tied to the data segment



System Engineering:
An Essential Part of Pan-STARRS' Success

- Coordination and integration
- Multiple technical disciplines – "orbits"
- Multiple phases of development
  - System requirements analysis and design
  - System implementation
  - System testing



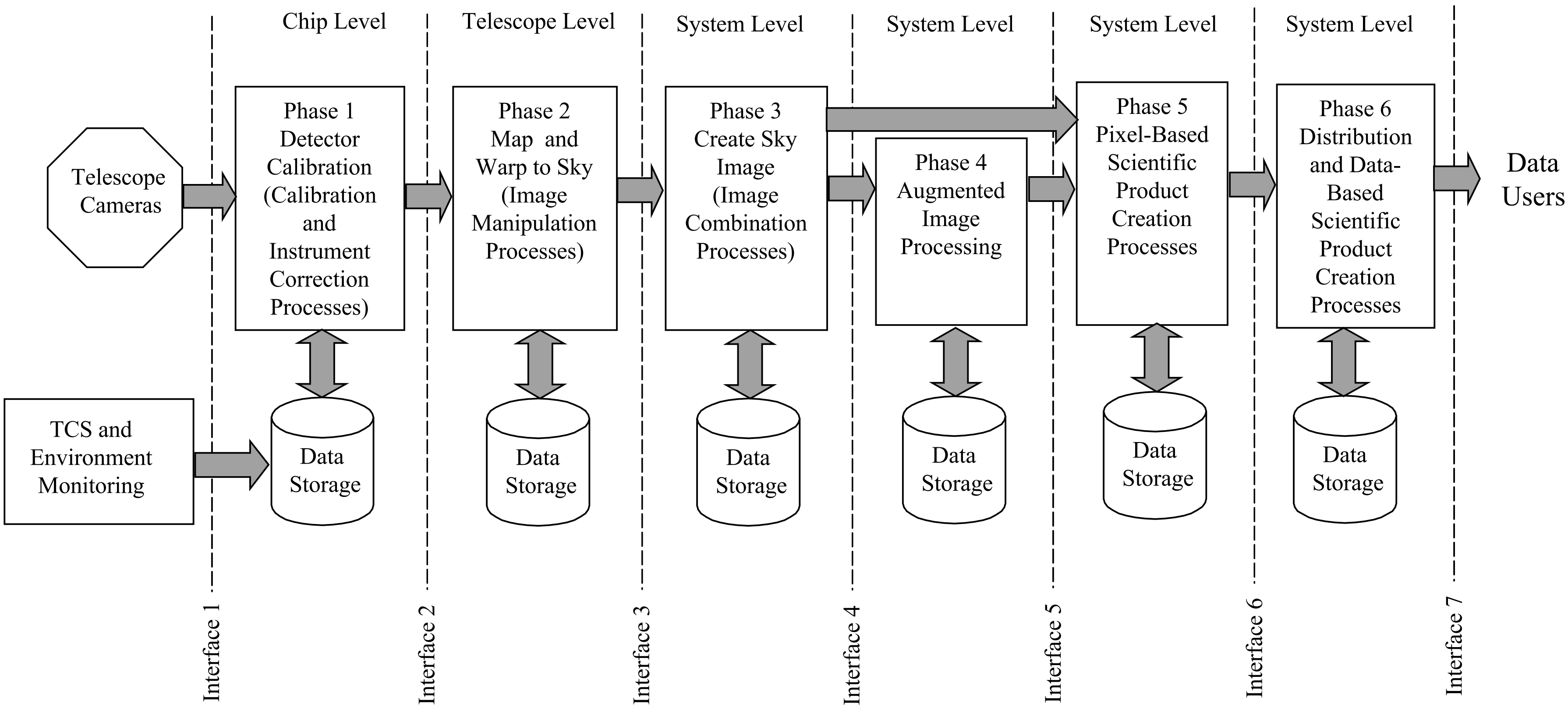
Data Segment: Technical Approach

Consider the whole system, not simply data objects and methods

- Remember the user: science requirements drive what is acquired, processed, stored, AND retrieved
- Data analysis, modeling, processing and management must be considered end-to-end, not separable components
- Data management is more than a database
  - Don't duplicate data: processing vs long-term storage
  - Management of data retrieval is essential
- Must consider long-term ramifications of very large data base management: technology, cost, science benefit
- Design to cost
  - Nominal vs maximal set of science requirements
  - Consider impact on upgrades of very large database → modular
  - Consider impact of new/improved processing algorithms → plug-able methods

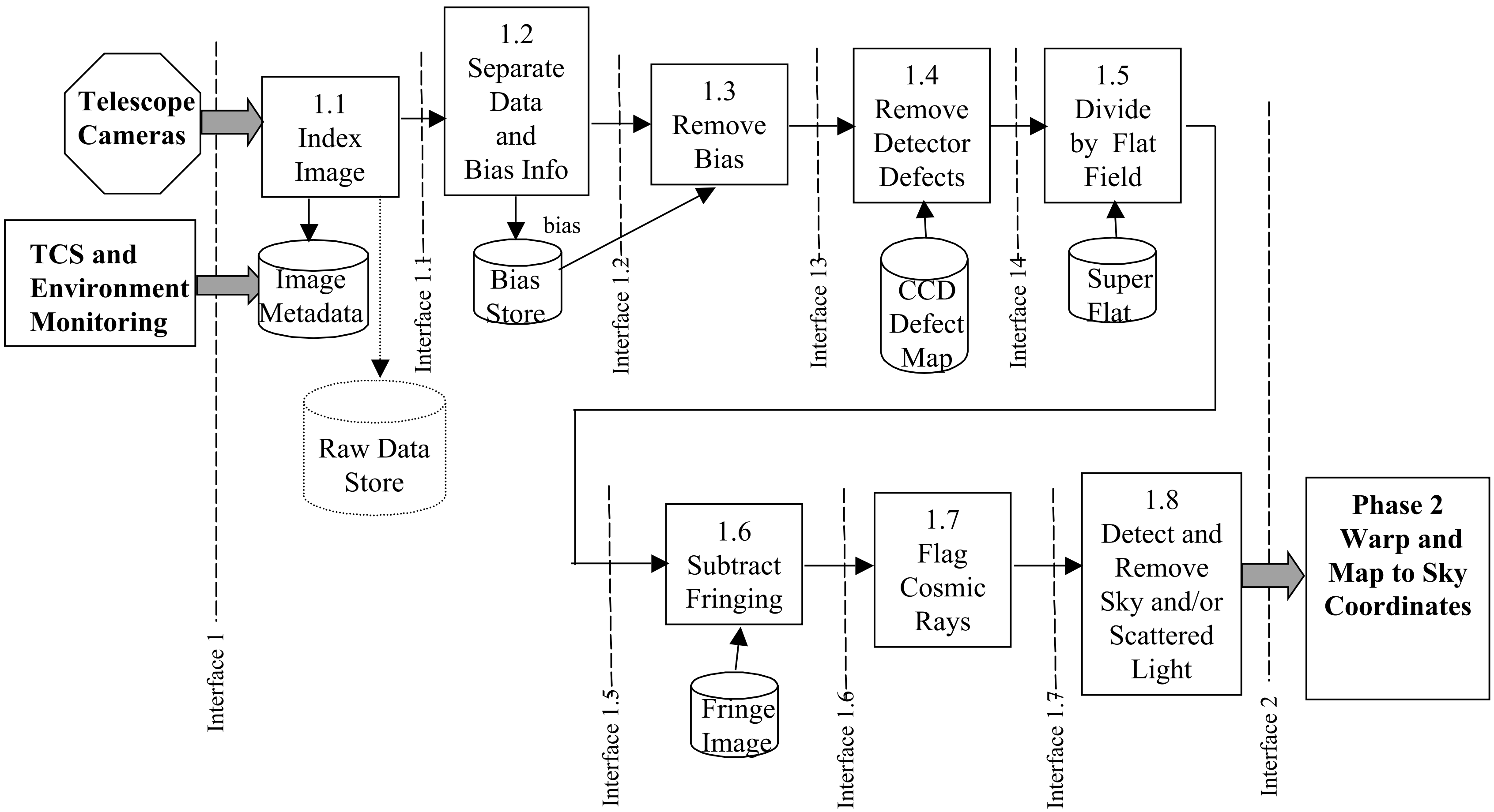
Current Concept of Data Flow: End-to-End

Top Level Data Flow and Interfaces



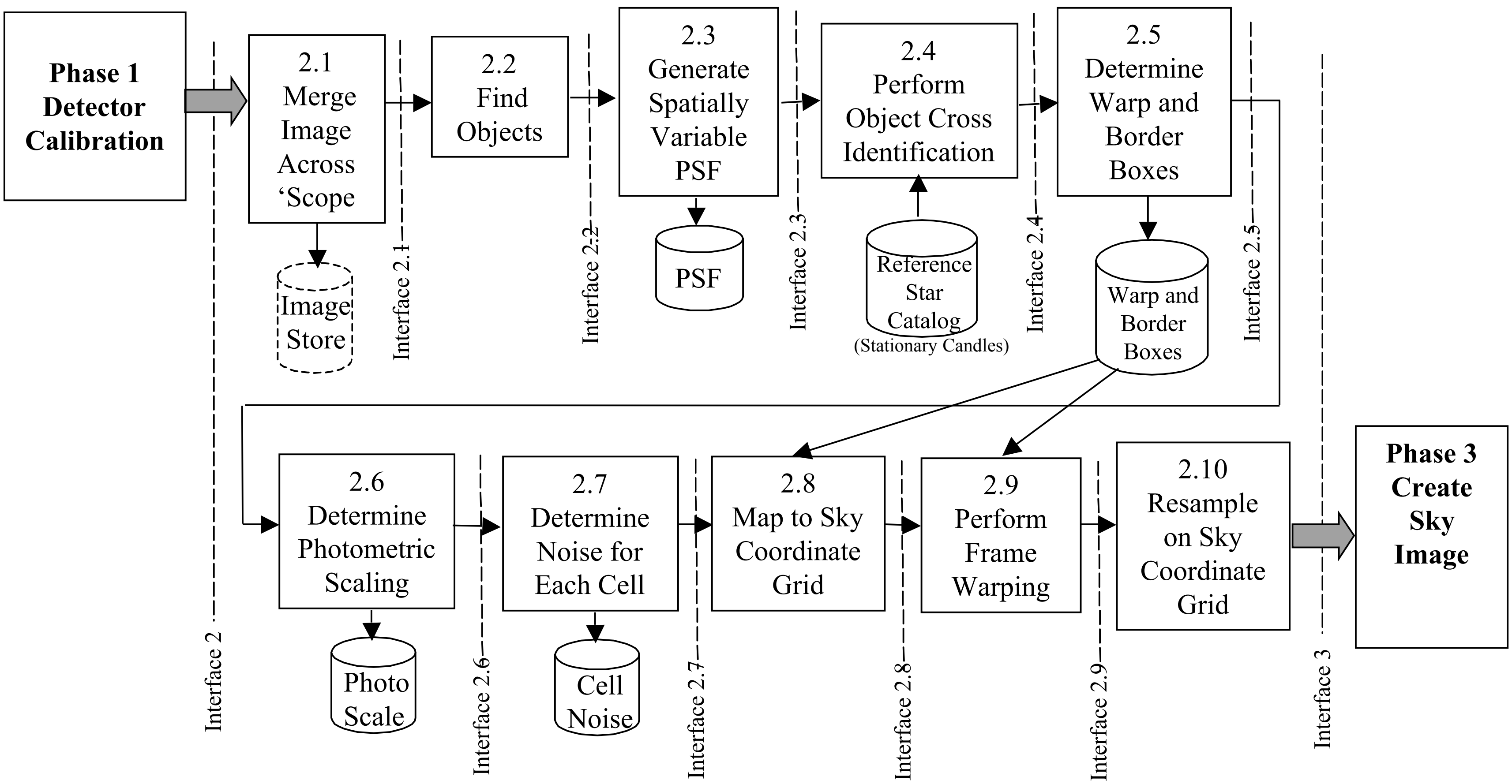
Second Level: Significant Phases

Phase 1: Detector Calibration



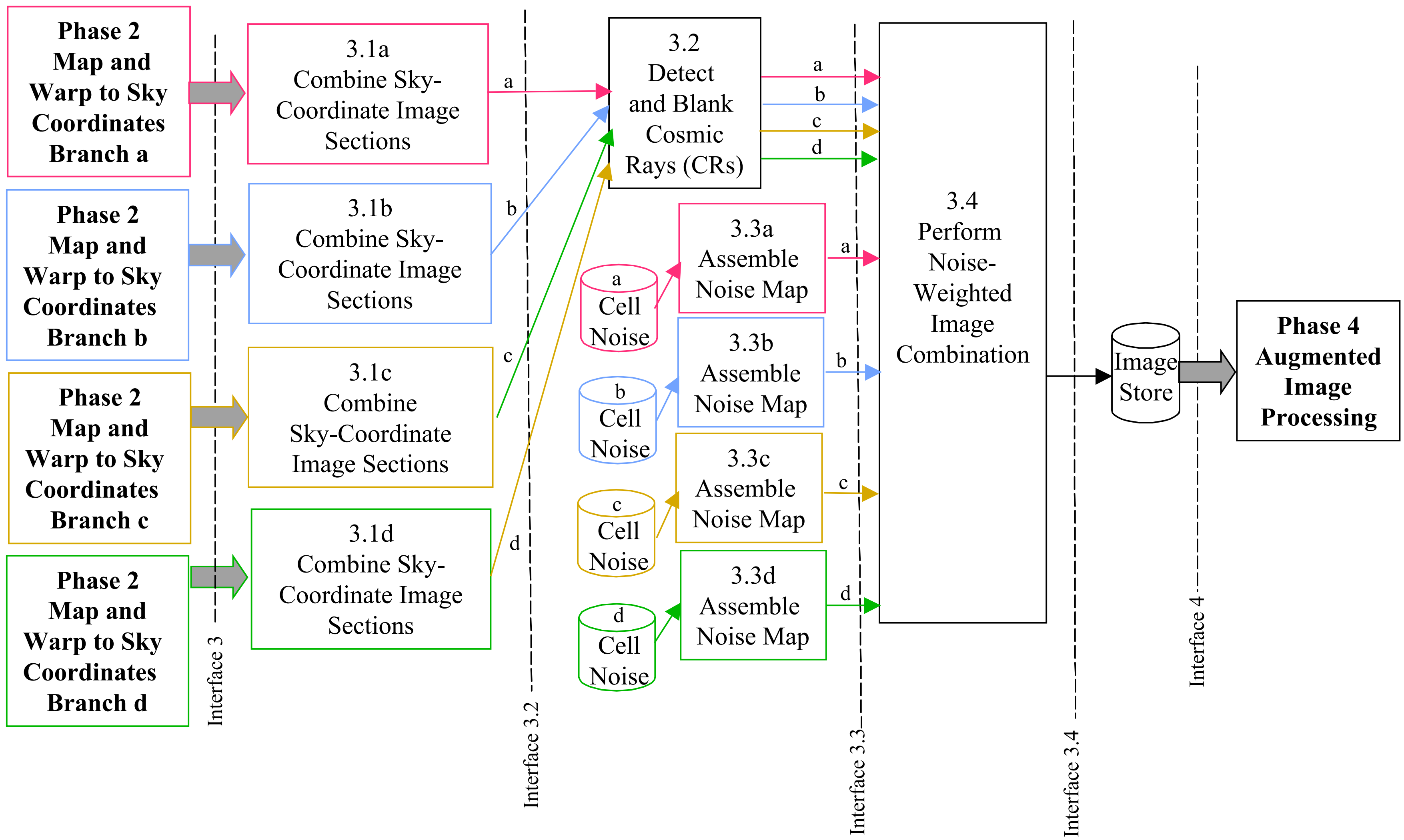
Telescope-centric coordinates, processes performed for each CCD

Phase 2: Map and Warp to Sky

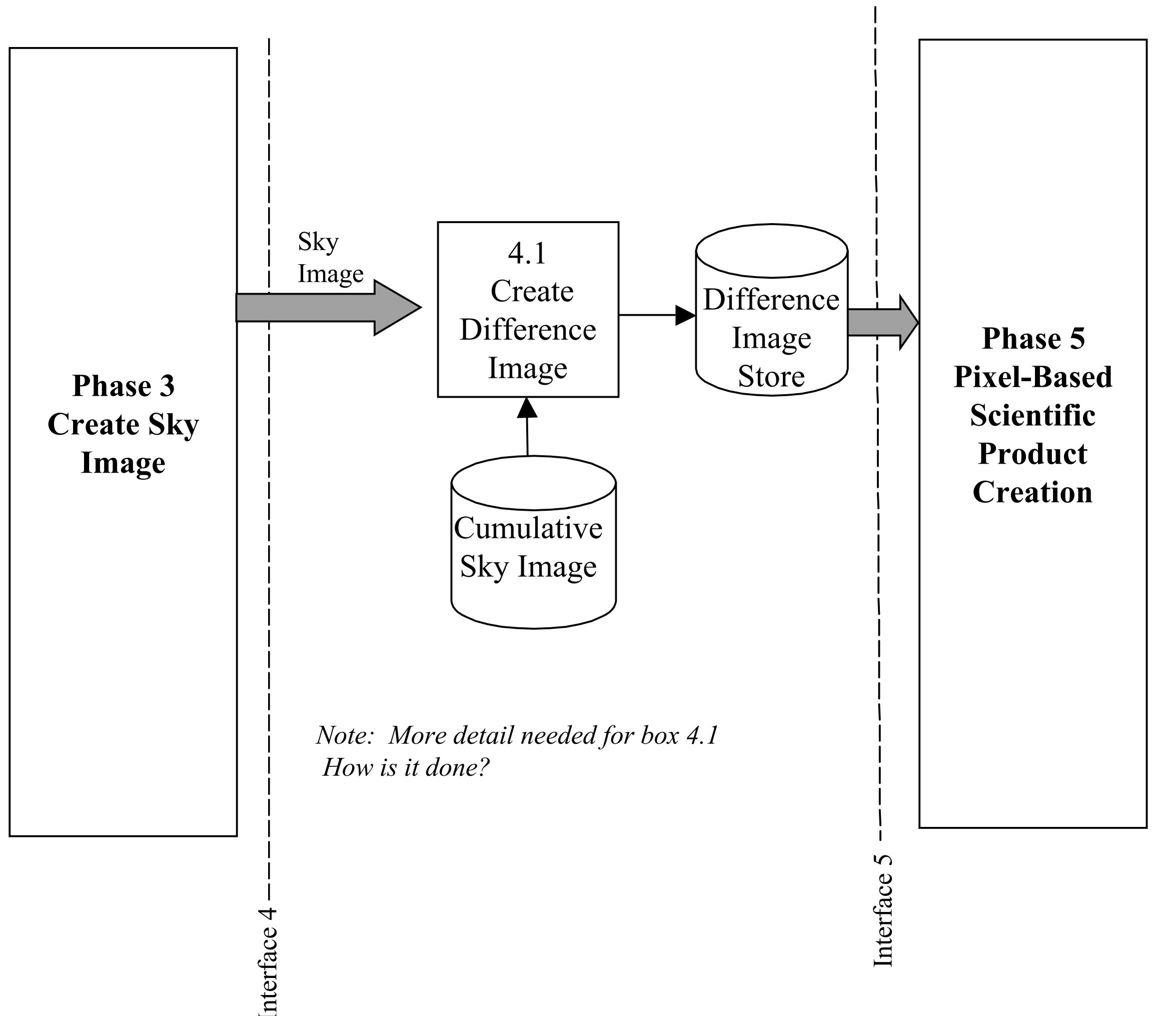


Telescope-centric coordinates, processes performed for each Telescope

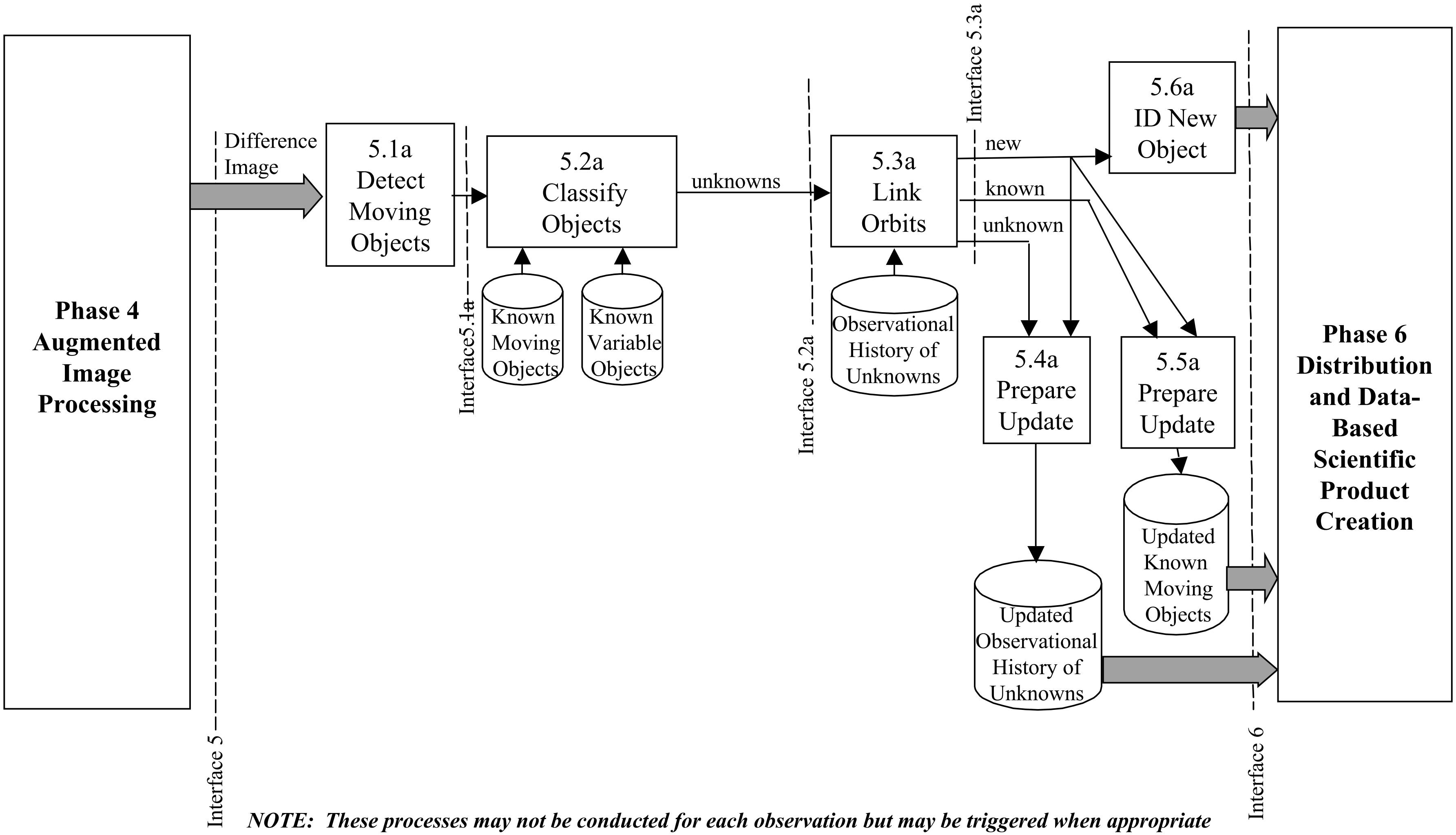
Phase 3: Create Sky Image



Phase 4: Augmented Image Processing

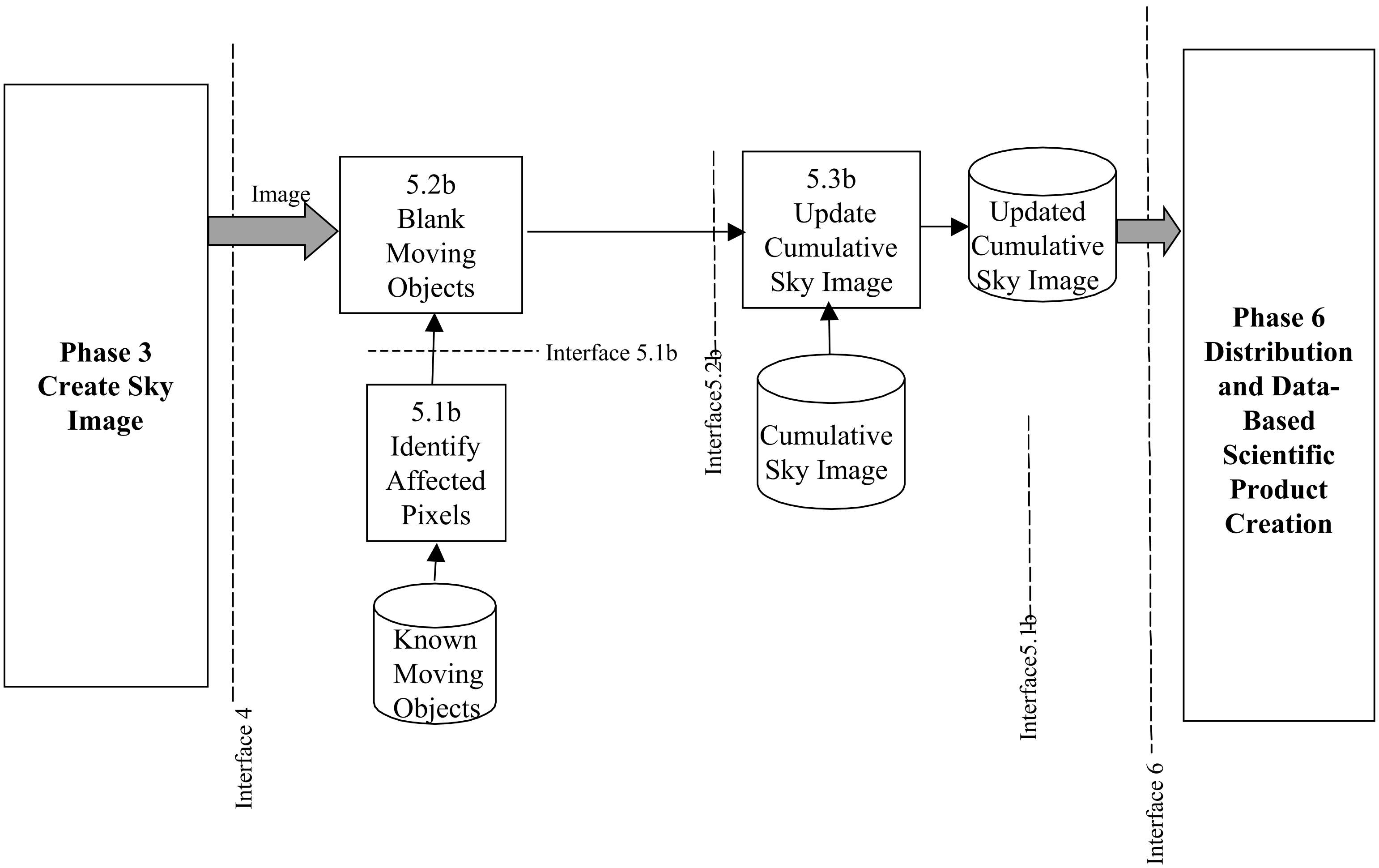


Phase 5A: Pixel-Based Scientific Product Creation-Basic – NEO Detection



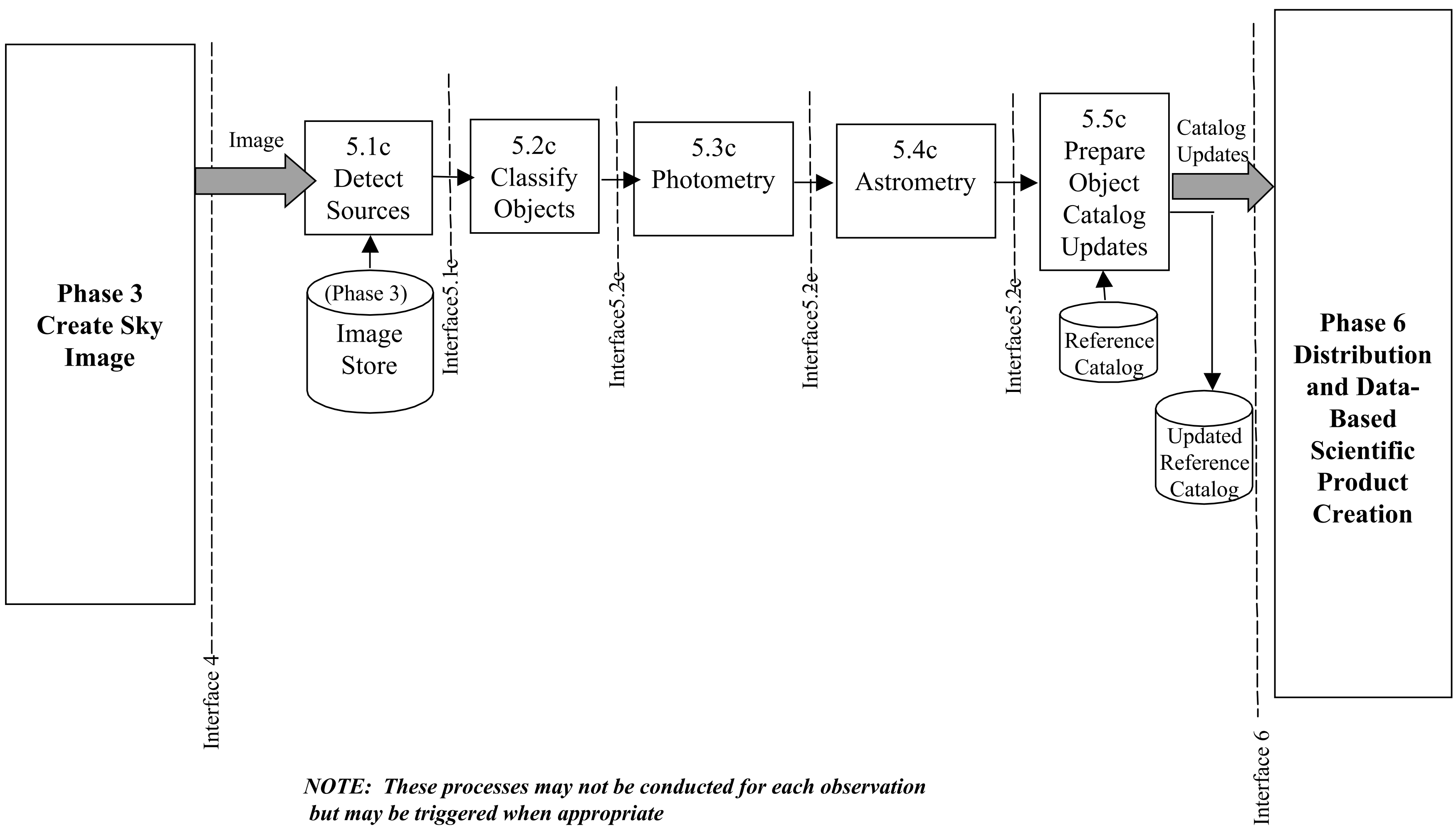
NOTE: These processes may not be conducted for each observation but may be triggered when appropriate

Phase 5B: Pixel-Based Scientific Product Creation-Basic – Cumulative Sky Image Update



NOTE: These processes may not be conducted for each observation but may be triggered when appropriate

Phase 5C: Pixel-Based Scientific Product Creation-Enhanced – Catalog Update



NOTE: These processes may not be conducted for each observation but may be triggered when appropriate

Phase 5D: Pixel-Based Scientific Product Creation-Enhanced – KBO Detection

Phase 4 Augmented Image Processing

Interface 5

TBD

Phase 6 Distribution and Data-Based Scientific Product Creation

Interface 6

NOTE: These processes may not be conducted for each observation but may be triggered when appropriate

Phase 5E: Pixel-Based Scientific Product Creation-Enhanced – Full Color Sky Image

University of Hawaii Astronomy

SAIC Science Applications International Corporation An Employee-Owned Company

Phase 4 Augmented Image Processing

TBD

Phase 6 Distribution and Data-Based Scientific Product Creation

Interface 5

NOTE: These processes may not be conducted for each observation but may be triggered when appropriate

Interface 6

Phase 5F: Pixel-Based Scientific Product Creation-Enhanced – Variable Static Objects

Phase 4 Augmented Image Processing

TBD

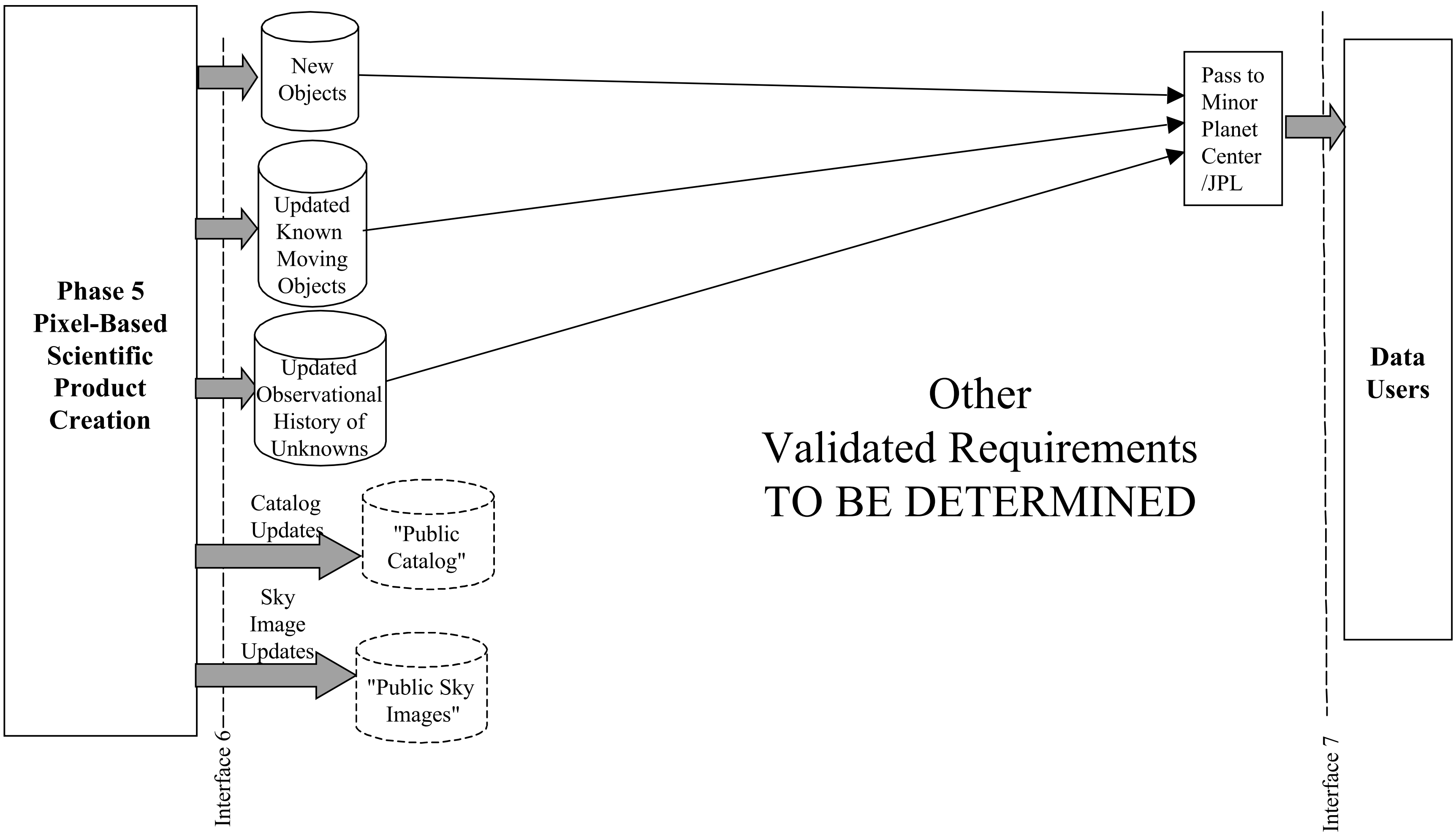
Phase 6 Distribution and Data-Based Scientific Product Creation

Interface 5

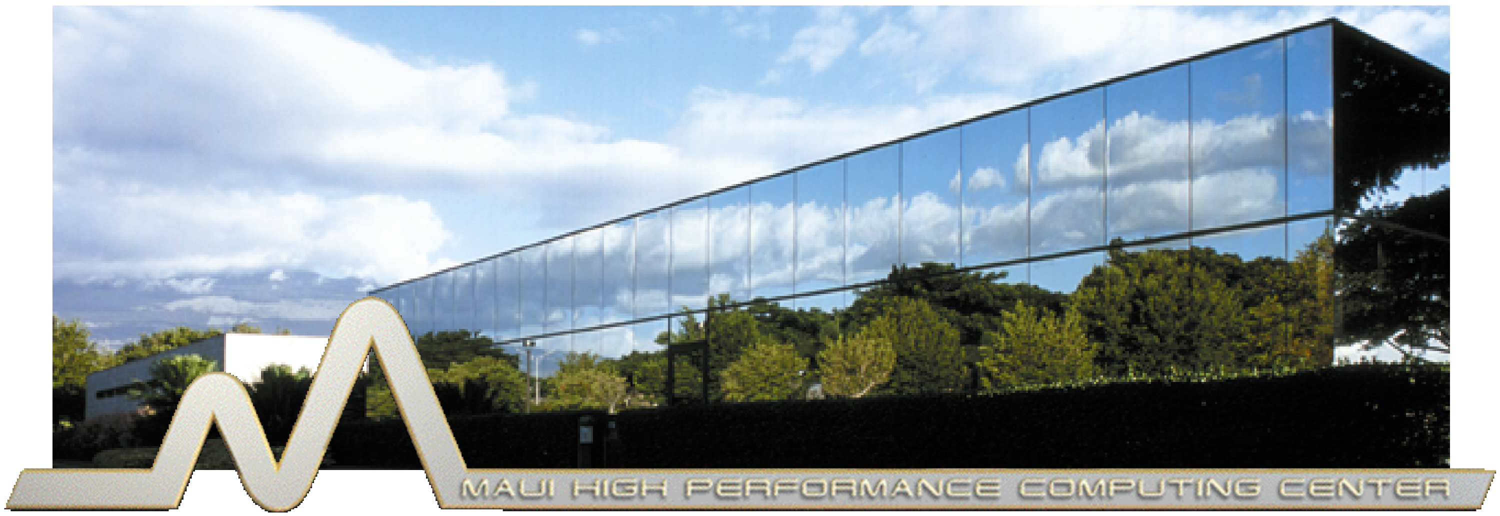
NOTE: These processes may not be conducted for each observation but may be triggered when appropriate

Interface 6

Phase 6: Distribution and Data-Based Scientific Product Creation



Data Processing – MHPCC



Data Processing - Topics

- Who are we?
- Implementation plan
- Risks and mitigations

Data Management – SAIC/Winter

SAIC Science Applications International Corporation
An Employee-Owned Company

Winter Corporation

Data Management - Topics

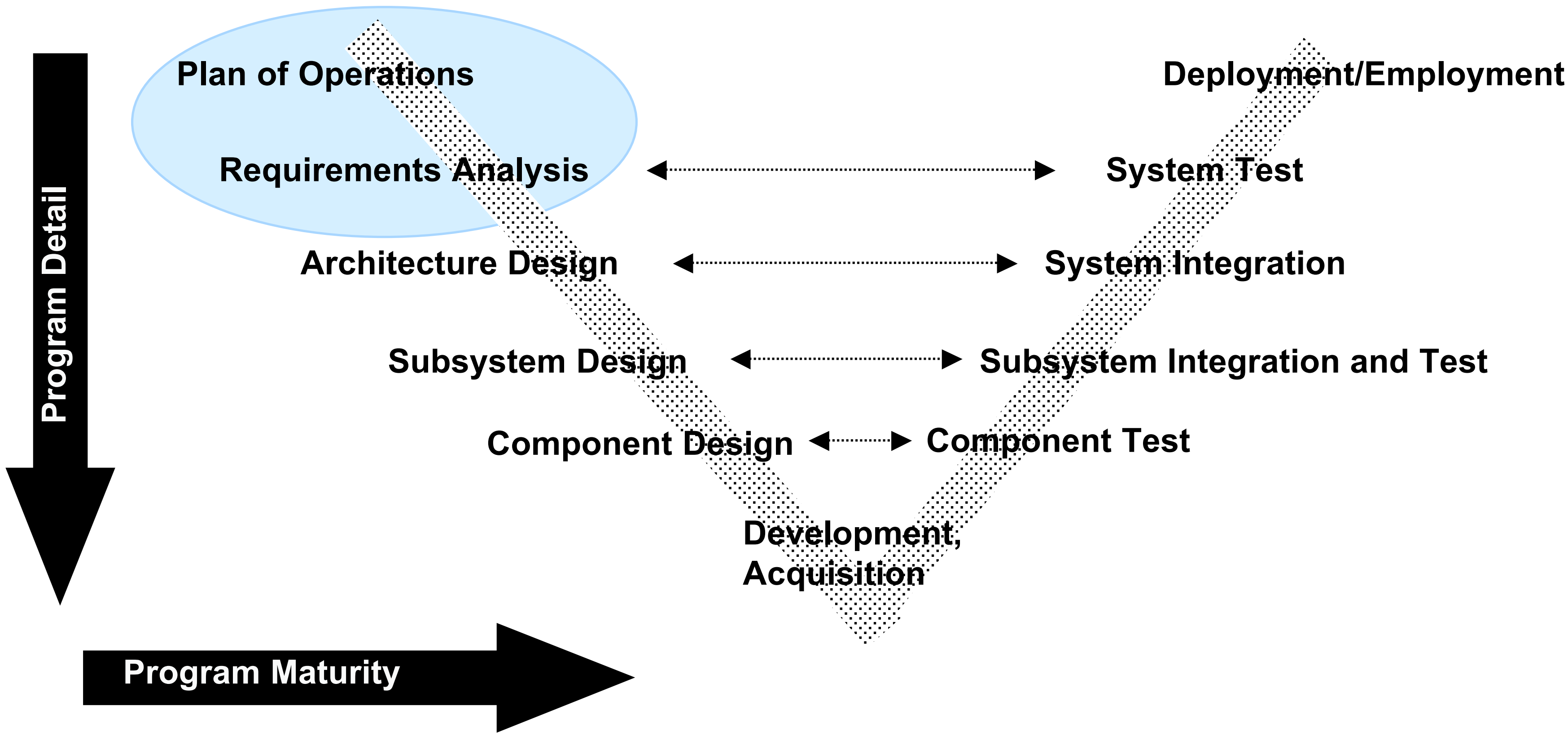
- Who are we?
- Implementation plan
- Risks and mitigations

Data Management - Who Are We?

- Dr. Julie Rosen (PI) – Mathematician, requirements analysis
- Dr. John Wick – Systems engineer, requirements analysis
- Dr. Richard Meyer – Systems architect, very large dbms analysis and design
- Mr. Wayne Smith – Systems engineer, system software analysis and design
- Dr. Robert Eek – Data architect, very large dbms design and implementation
- Dr. Mark Mekaru – SAIC Sr. Vice President, engineering

Name	Phone	Email
Dr. Julie Rosen	703-676-7354	Julie.A.Rosen@saic.com
Dr. John Wick	703-676-2738	John.A.Wick@saic.com
Dr. Richard Meyer	703-941-7526	Richard.Meyer@mindspring.com
Mr. Wayne Smith	703-676-6522	Wayne.L.Smith@saic.com
Dr. Robert Eek	703-676-5776	Robert.E.Eek@saic.com
Dr. Mark Mekaru	703-248-7723	Mark.M.Mekaru@saic.com

SAIC's Systems Engineering Practices: Essential for Effective Implementation



All requirements must be testable and tested

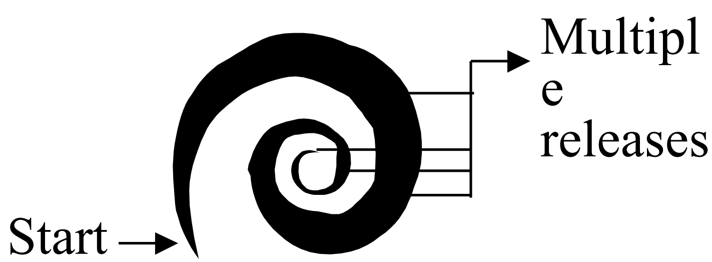
Year 1

SAIC's Structured Software Development

- Process
  - Rigorous analysis of *all* requirements
    - Functional Requirements
    - Performance Requirements
    - Maintenance Requirements
    - Derived Requirements
  - Development and testing to ensure compliance with *all* requirements
  - Automated configuration control – source code and requirements
  - Adherence to software development standards ensures consistency ➔ comprehension ➔ reduced development/maintenance efforts ➔ reduced cost and schedule
- Benefits
  - No surprises after deployment
  - Increased reliability and maintainability translate directly to reduction in overall system cost

Spiral Approach To Data Management Implementation

- Build a little, test a little, then iterate
- Applicable for dynamic problems
  - Basic system requirements known, but details of data flow are in flux at the start
  - System and data requirements will be refined with each iteration, but consistent with basic data management architecture
  - Data processing/management component development are autonomous, provided interface is defined, and collaboration in early design
- System component development done in parallel provide feedback to data management
  - Detectors output influence data processing performance
  - Pipeline processing algorithms impact performance requirements of data management
  - Data management/storage technology (cost vs performance) affects feasibility of meeting science requirements



Data Management: Implementation Plan (1 of 2)

- Year 1 – Develop concept of operations
  - Systems-wide "holistic" approach requires collaboration with science, data processing, and systems engineering staff to develop data flow requirements
  - Principal users, their data needs
  - Trade-off studies: cost versus science benefit
- Year 2: Prototype critical development components/ end-to-end path
  - Database architecture with fully loaded contents (simulated data) gives performance bounds, better hardware/firmware/software specs
  - Basic database architecture, simulate/prototype telescope "unit," simulate/prototype data feed, simulate/prototype data storage procedure (including estimate of manual load)
  - Determine trade-off parameters, thresholds, interface details, improved cost estimates, need for enhancements
  - Revise/refine system requirements: analysis, design, interface specs

Data Management: Implementation Plan (2 of 2)

- Year 3: Development and Test
  - Identify component layers and implementation phases
  - Parallel development of components
    - Include periodic testing and feedback for later iterations' inclusion
    - Refinement of interface specification among components
    - High and detailed design of component's modules: I/O among modules, algorithms within modules, I/O with other components
    - Implementation and component testing using simulated I/O data
    - Constant use of configuration control
  - Integration of components
    - Hardware-to-hardware, hardware-to-software, software-to-software
    - Implementers perform testing
    - Results fed-back for inclusion in next iteration
  - System testing
    - Non-implementer testing, as appropriate for component
    - Results used to refine loading, usability issues
- Year 4: Production and maintenance
  - Repeat development cycle as needed

Areas of Technical Risk for Data Management

- Completeness of design
  - Different access/retrieval methods mandate distinct data structure/storage design
  - High speed buffering required at multiple points to support both re-start and intermediate results
- Performance
  - Processing latency must meet science requirements needs
  - Data storage must be designed to meet retrieval requirements
- Scaling
  - Large volume of image files increases cost, power, floor space – beyond budget?
  - Technology dead-ends may preclude full implementation of science requirements
- Evolution
  - Dead-end design can't be upgraded to meet future requirements
  - Advances in technology cannot be incorporated effectively

Mitigating the Data Management Risk

Risk Area	Mitigation Approach (as seen from year 1)
Completeness of design	▪ Thorough understanding of science and processing requirements ▪ Capture data flow for the whole system
Performance	▪ Compare design against existing comparables ▪ Prototype elements of concern
Scaling	▪ Conduct trade studies ▪ Prototype elements of concern
Evolution	▪ Identify decision points ▪ Maintain upgrade strategies/alternate paths