NASA AMES
Virtual Airspace Modeling and Simulation (VAMS)

Air Traffic Management System Development & Integration (ATMSDI)

VAMS TIM #2

Airspace Concepts Evaluation System: Overview

Douglas Sweet

28 August 2002

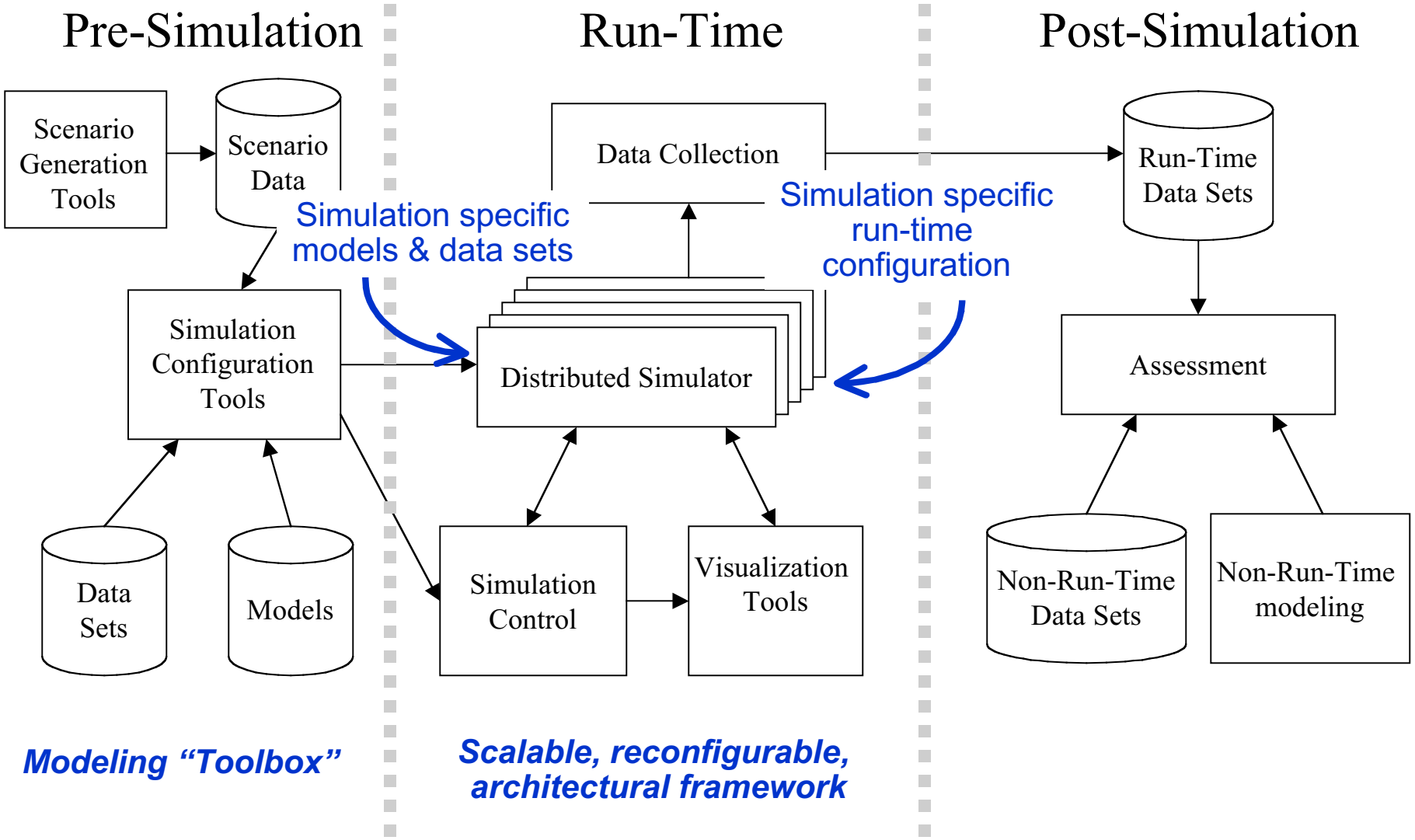
Outline

- **Overview**
  - System
  - Modeling
  - Architecture
- **Prototype System**
- **Build 1 System**

ACES Requirements

- **Represent interdependencies among NAS participants**
  - Current and future NAS is highly interactive
  - Requires NAS-wide simulation
- **Represent a wide variety of operational concepts**
  - New systems, new roles / responsibilities
  - Requires adaptable and flexible system
- **Provide broad assessment capabilities**
  - Operational, economic, and safety metrics
  - Requires models capable of producing a wide range of data
- **Provide a practical implementation approach**
  - Ease in developing and running a simulation
  - Ease in integrating new capabilities
  - Efficient use of computational resources
  - Requires tailored simulations using varying degrees of model fidelity

Airspace Concept Evaluation System

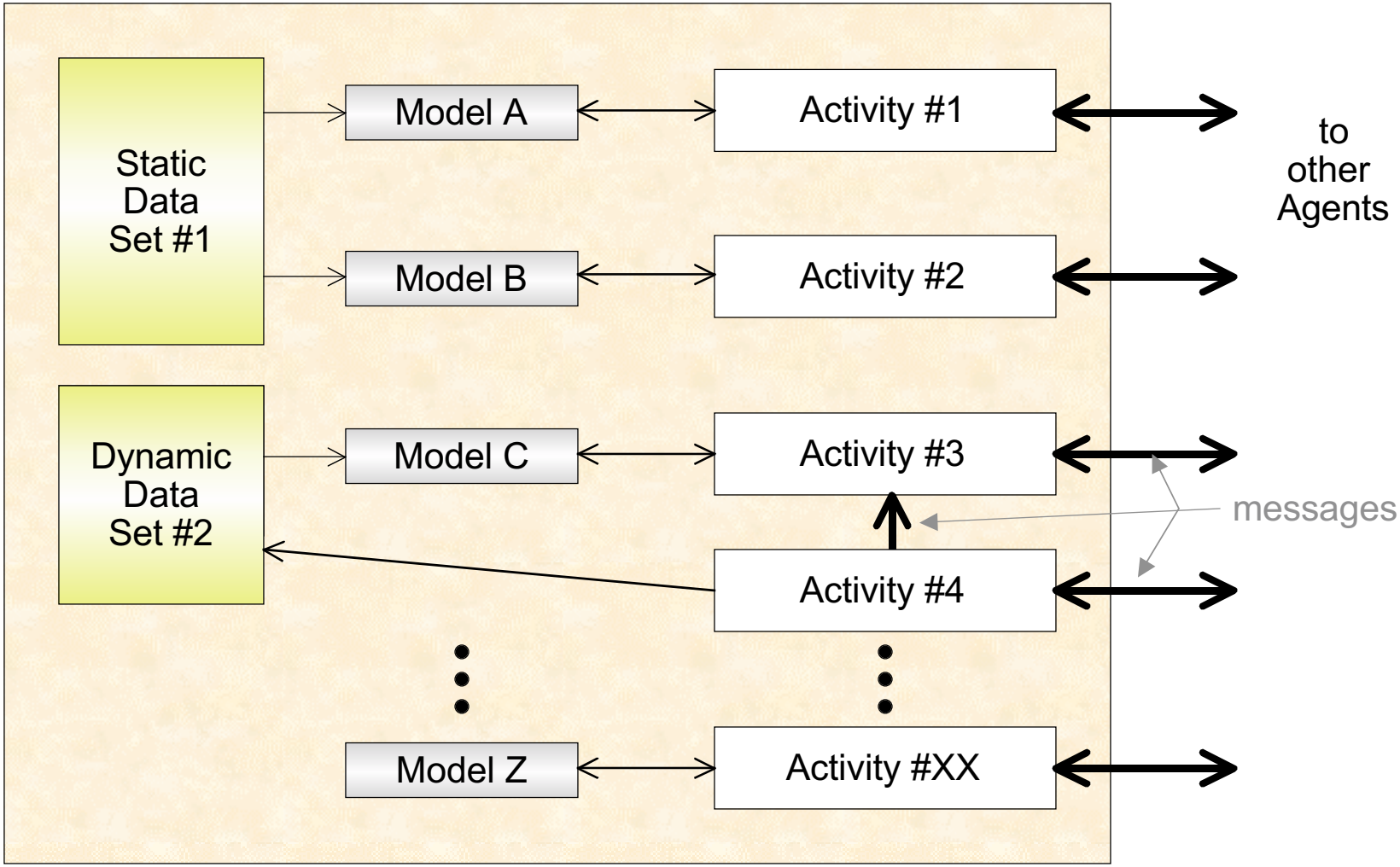


ACES Core Modeling Approach

- **Agent-based paradigm:**
  - Object oriented
  - each Agent made up of activities
  - each activity supported by individual models
  - Communication by messages
- **One-to-one correspondence to the NAS:**
  - Agents ⇒ NAS participants / entities
  - Activities ⇒ NAS participant's functions
  - Messages ⇒ NAS CNS systems
  - Data Sets ⇒ NAS environment
- **Multiple levels of model fidelity available**

Agent Example

Agent X



ACES Core Modeling Approach

- **Agent examples:**
  - Aircraft: Flights, Pilots
  - Airline: Dispatchers, Ramp Managers
  - Air Traffic Control: ATCSCC, Sector Controllers, Traffic Management Units (TMU)

- **Activity examples:**
  - Flight: trajectory propagation, TCAS, Flight Management System
  - ATCSCC: Monitor Alert, Ground Delay Program, Ground Stop Program
  - Sector Controller: voice communications, conflict detection, conflict resolution, flight plan updates, hand-offs

ACES Core Modeling Approach

- **Agent to Agent message examples:**
  - Flight to Controller voice communication
  - Flight to Controller aircraft state data (radar-based)
  - Flight to Flight to Controller aircraft state and intent data (ADS-B)

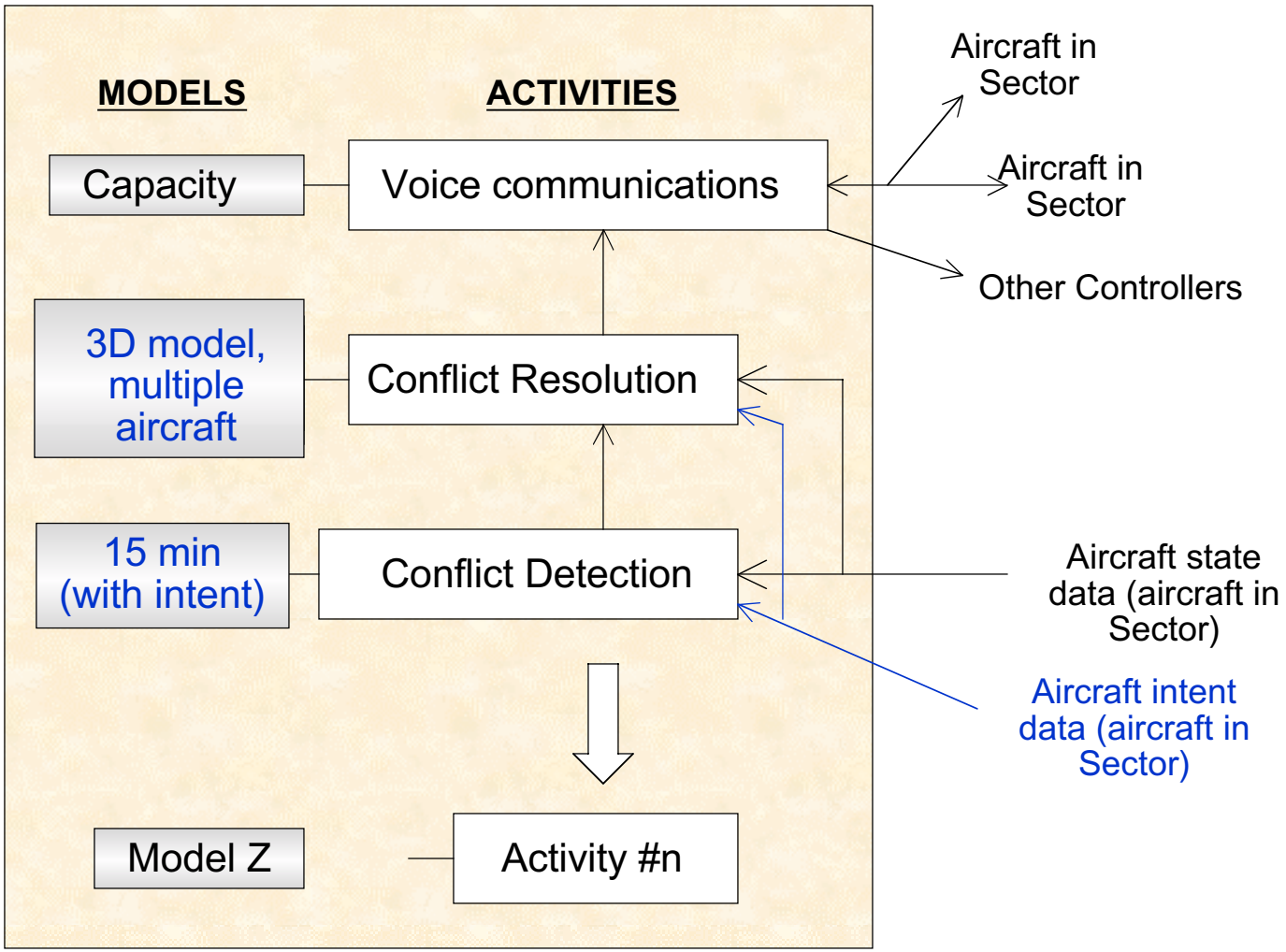
- **Agent data set examples:**
  - Static (Airport locations, airspace definitions, facility boundaries)
  - Dynamic (winds, convective weather, dynamic facility boundaries)

ACES Core Modeling Approach

- **Multiple levels of model fidelity: examples**
  - **Flight trajectory propagation model**
    - » High fidelity - 4 DOF force model
    - » Medium fidelity - 3 DOF kinetic model
    - » Low fidelity - instantaneous acceleration
  - **Flight management system**
    - » High fidelity - FMS emulator
    - » Medium fidelity - airspeed, altitude, and route deviations
    - » Low fidelity - no trajectory deviation

Agent Example

Sector Controller Agent



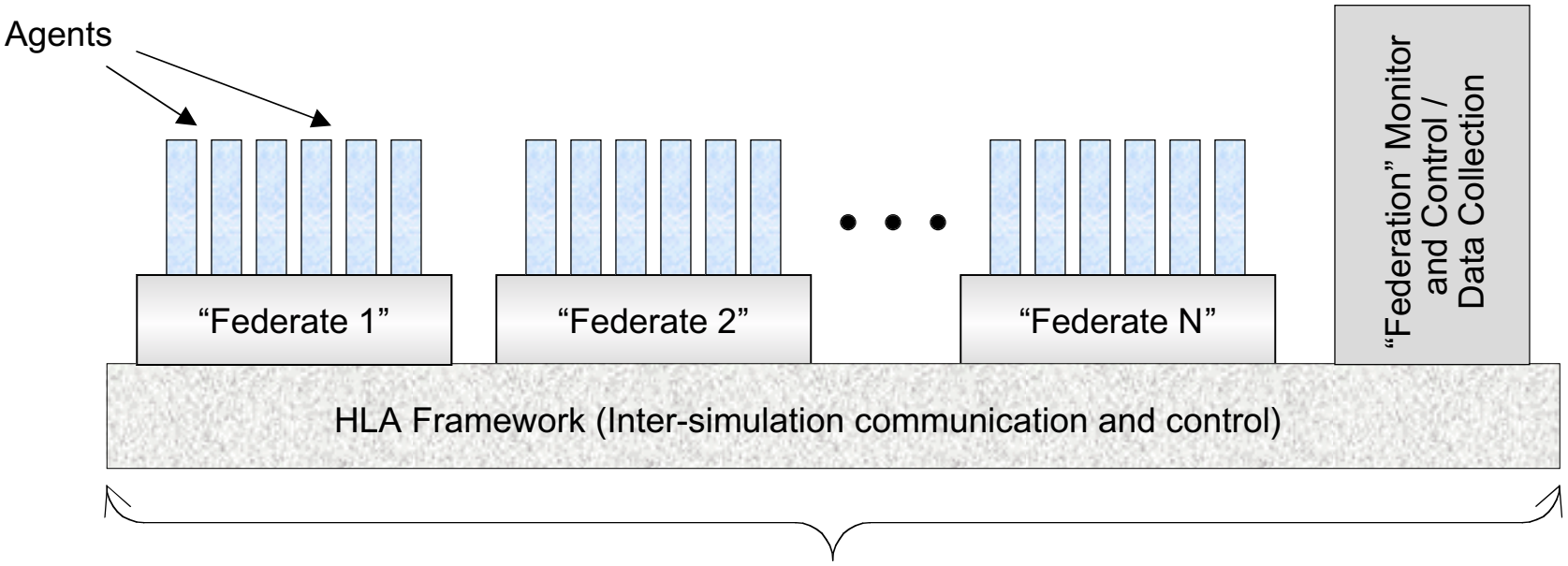
ACES Core Modeling Approach

- **Benefits**
  - One-to-one correspondence with NAS provides ability to isolate functionality
  - Modularity supports integration of new concepts
  - Supports flexibility in allocating Agents across the ACES distributed simulation framework

ACES Architectural Approach

Utilize High Level Architecture (HLA)

- Proven framework for large, distributed simulation
- Open architecture, widely used and supported
- Flexible and expandable



Federation Object Model (FOM) - specifies communication protocol between federates

Run-Time Infrastructure (RTI) - a communications infrastructure for federate to federate communication services

ACES Architectural Approach

Utilize an agent-based modeling and simulation engine:

Software layer between the agents and the HLA RTI

Supports intra-federate and inter-federate agent communication

Provides a well-defined modeling interface independent of the HLA implementation

model development independent of specific implementation

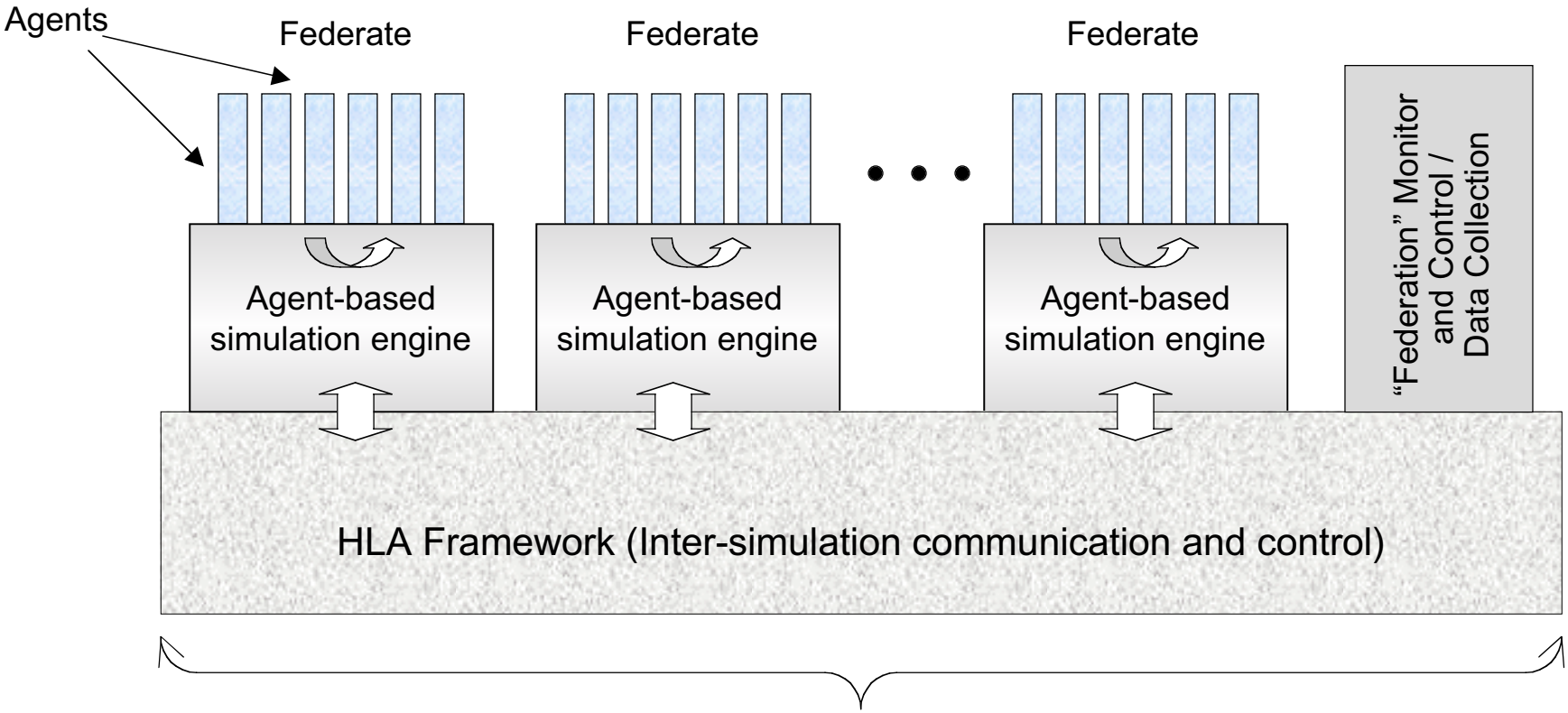
model development requires no knowledge of HLA

supports ease in allocating Agents for efficient utilization of computational and network resources

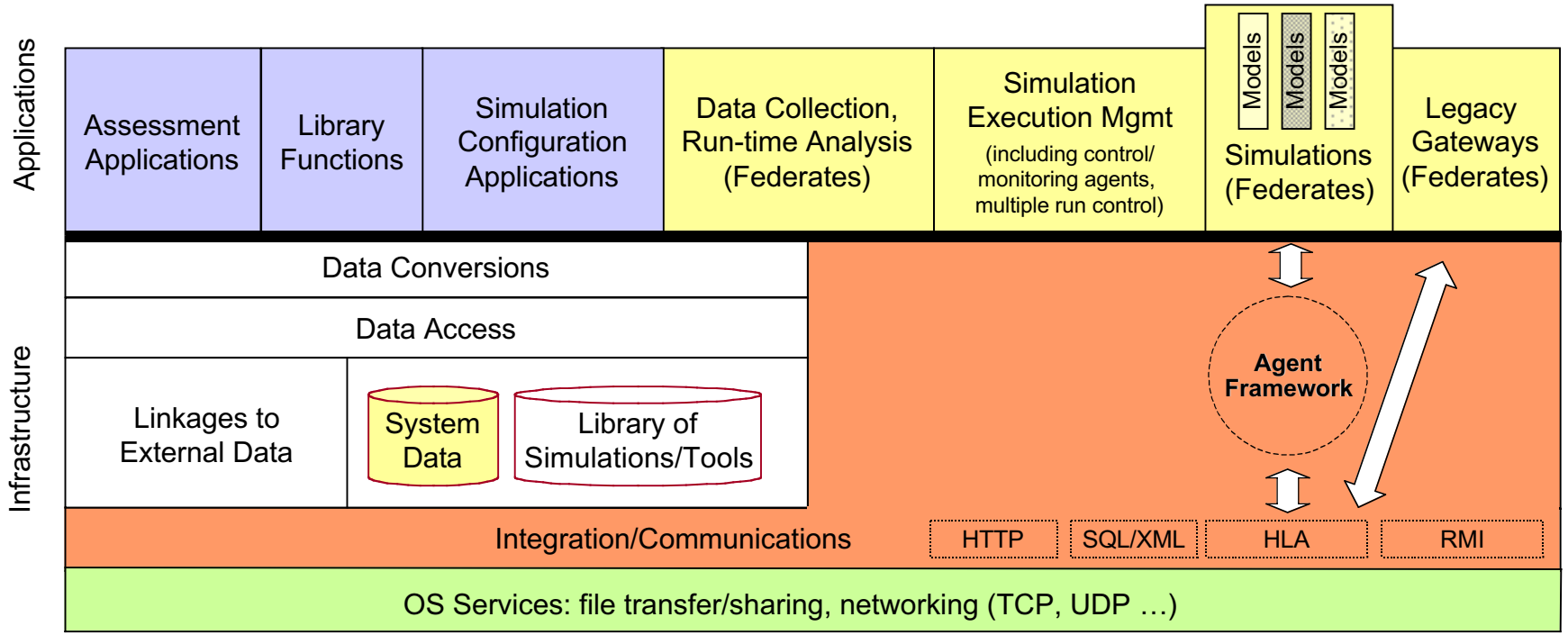
Provides a filtering mechanism to minimize HLA network traffic and improve overall performance

ACES Core Architectural Approach

- HLA provides Federate level flexibility / scalability
- Simulation Engine provides flexibility in allocation of Agents
- Allows the Agent to be the building block of the ACES simulation



ACES Overall Architecture



ACES Development

- **Prototype Demonstration System (completed)**
  - Demonstrate the use of HLA in a distributed, fast-time simulation
  - Demonstrate Agent-based modeling
- **Baseline System Development (in progress)**
  - Create a NAS-wide baseline simulation system
  - Validate the baseline system
- **System Enhancements**
  - Enhance Model Toolkit
  - Enhance architecture for performance / usability
  - Support VAMS concept evaluation and integration

ACES Development

- **Prototype Demonstration System (completed)**
  - Create a proof-of-concept system to demonstrate the use of HLA in a distributed, fast-time simulation
- **Baseline System Development (in progress)**
  - Create a NAS-wide baseline simulation system
  - Validate the baseline system
- **System Enhancements**
  - Enhance Model Toolkit
  - Enhance architecture for performance / usability
  - Support VAMS concept evaluation and integration

ACES Prototype Demonstration System

- **Transition agent-based, legacy simulation to HLA environment**
  - Utilize IAI's Agents-En-Route (AER)* NAS-wide simulation
  - Distribute AER agents into three separate "federates"
  - Integrate with HLA RTI, create FOM
- **Integrate centralized data collection and simulation control tools**
- **Extend modeling capabilities**
  - Incorporate "managed" aircraft paradigm (e.g. CD&R for sector controller, aircraft following an ETMS based flight plan)

. . . . . All in a four month period

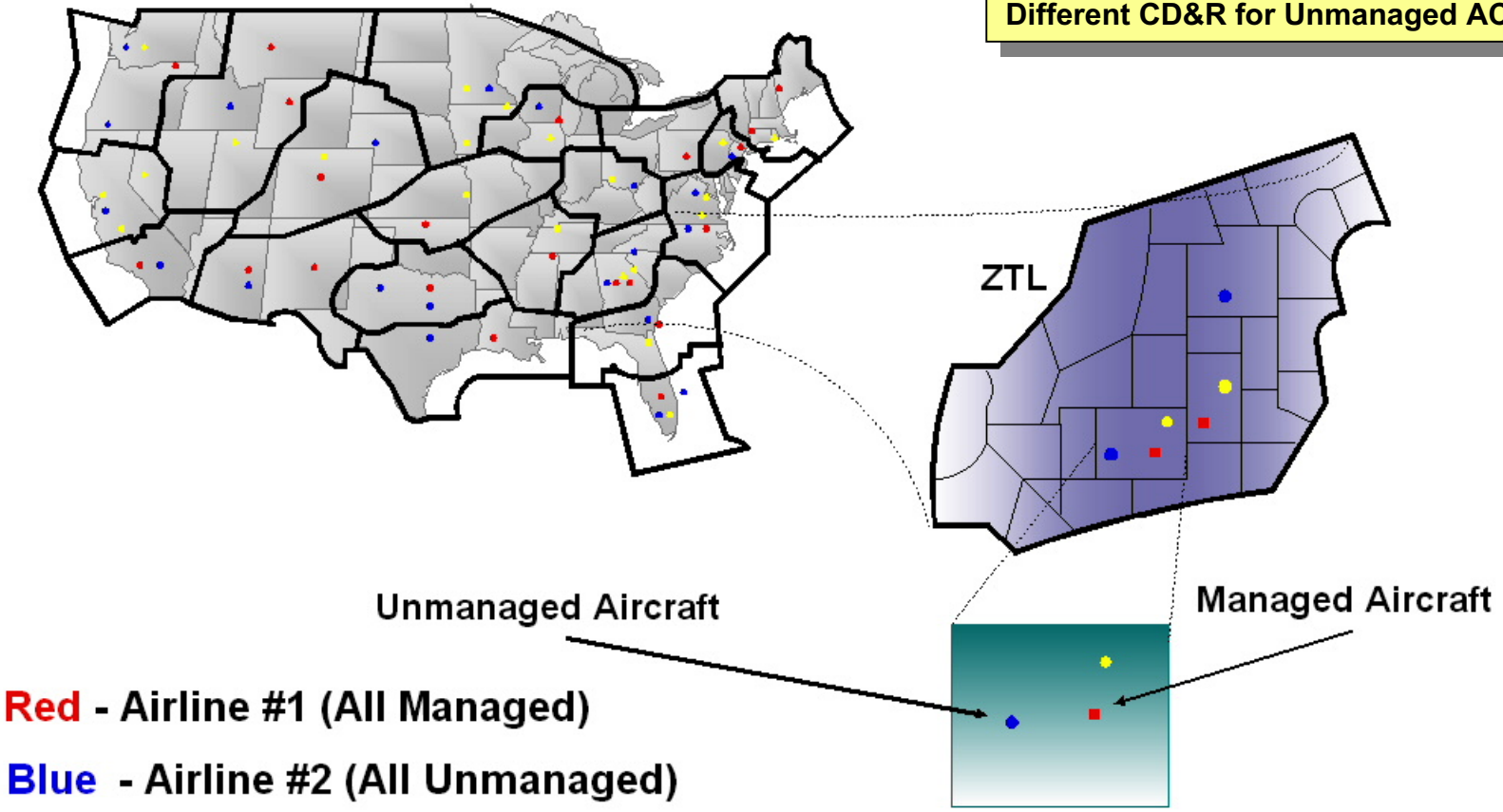
* developed under NASA SBIR, leveraging NASA's FACET simulation

Prototype: Simulation Description

NAS- Wide Enroute Simulation

Managed and Unmanaged AC in same airspace

Different CD&R for Unmanaged AC

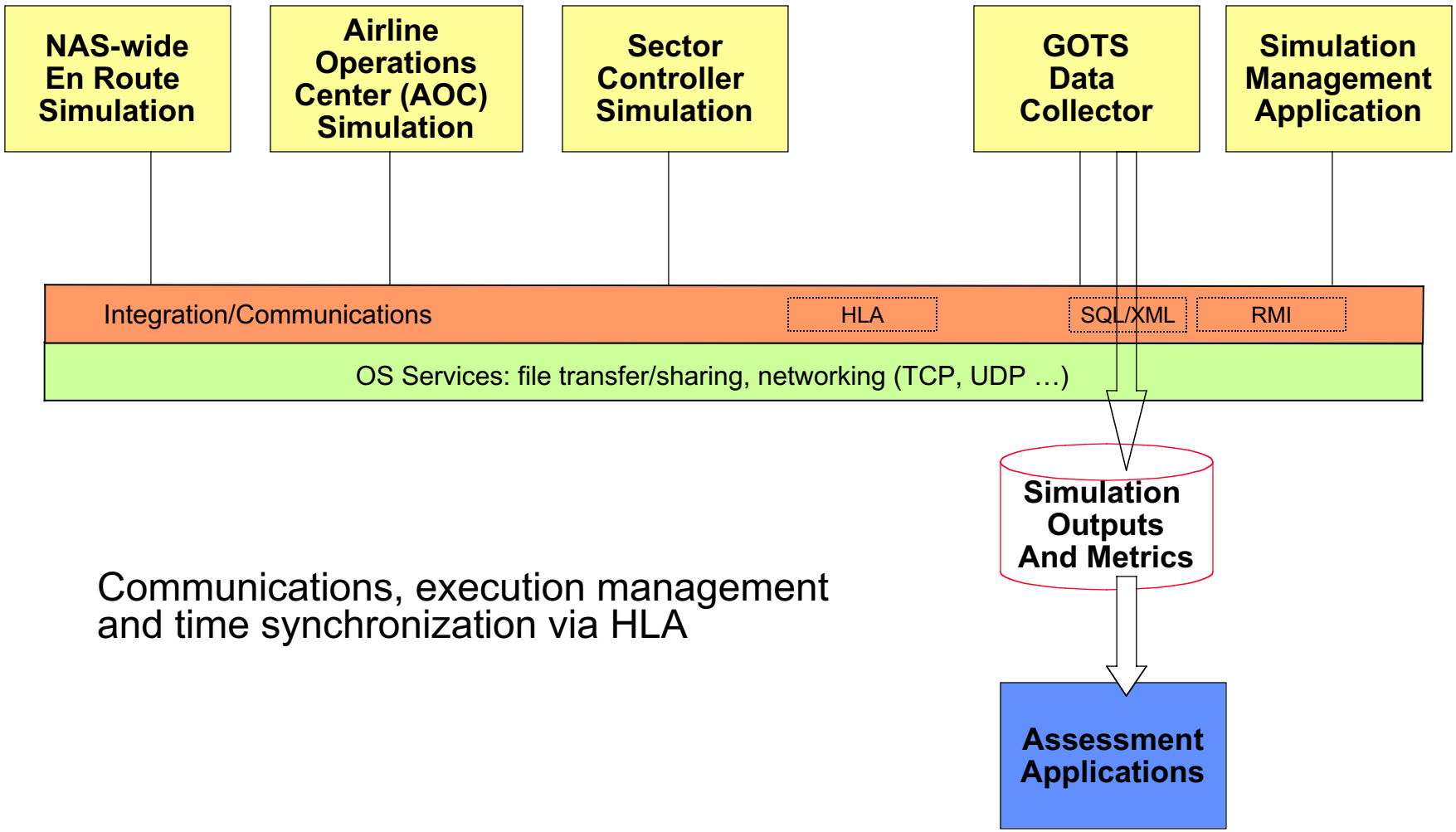


Red - Airline #1 (All Managed)

Blue - Airline #2 (All Unmanaged)

Yellow - Airline #3 (Mix Fleet)

Prototype Implementation



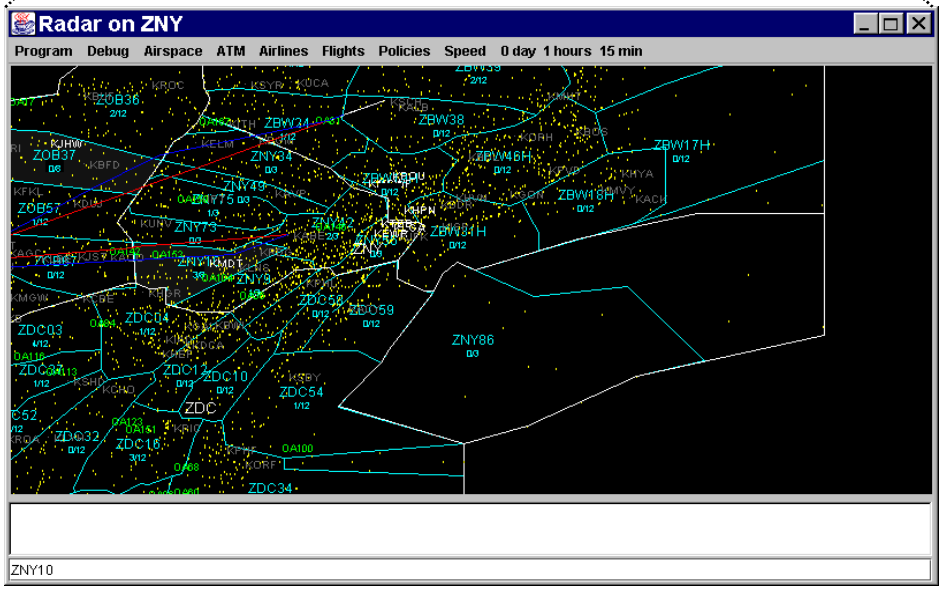
Communications, execution management and time synchronization via HLA

Prototype Configuration

Simulation Manager & Data Collection Federates

AER (En-Route Simulation) Federate

AOC & Controller Federates



Lessons Learned from Prototype

- **HLA-based architecture supported distributed simulations**
  - five interacting federates
- **Agent-based paradigm a good match for ACES**
  - provides clean interface to support efficient distribution of models in a distributed environment
  - supports ability to efficiently integrate new capabilities
- **Identified key needs for Build 1 system**
  - Need to incorporate HLA capabilities not utilized in prototype for improved performance
  - Need to support ease of model integration - Prototype modeler needed to understand HLA
  - Need to create foundation for ACES

ACES Build 1 System

- **Provide the architectural foundation**
  - **Create an agent infrastructure**
    - » modeler independence from HLA
    - » improved efficiency
    - » ease of reconfiguration
  - **Develop a robust HLA framework**
    - » ground up design for large scale simulation
    - » address key design issues (repeatability, time management)
    - » simulation initialization
    - » simulation configuration

ACES Build 1 System

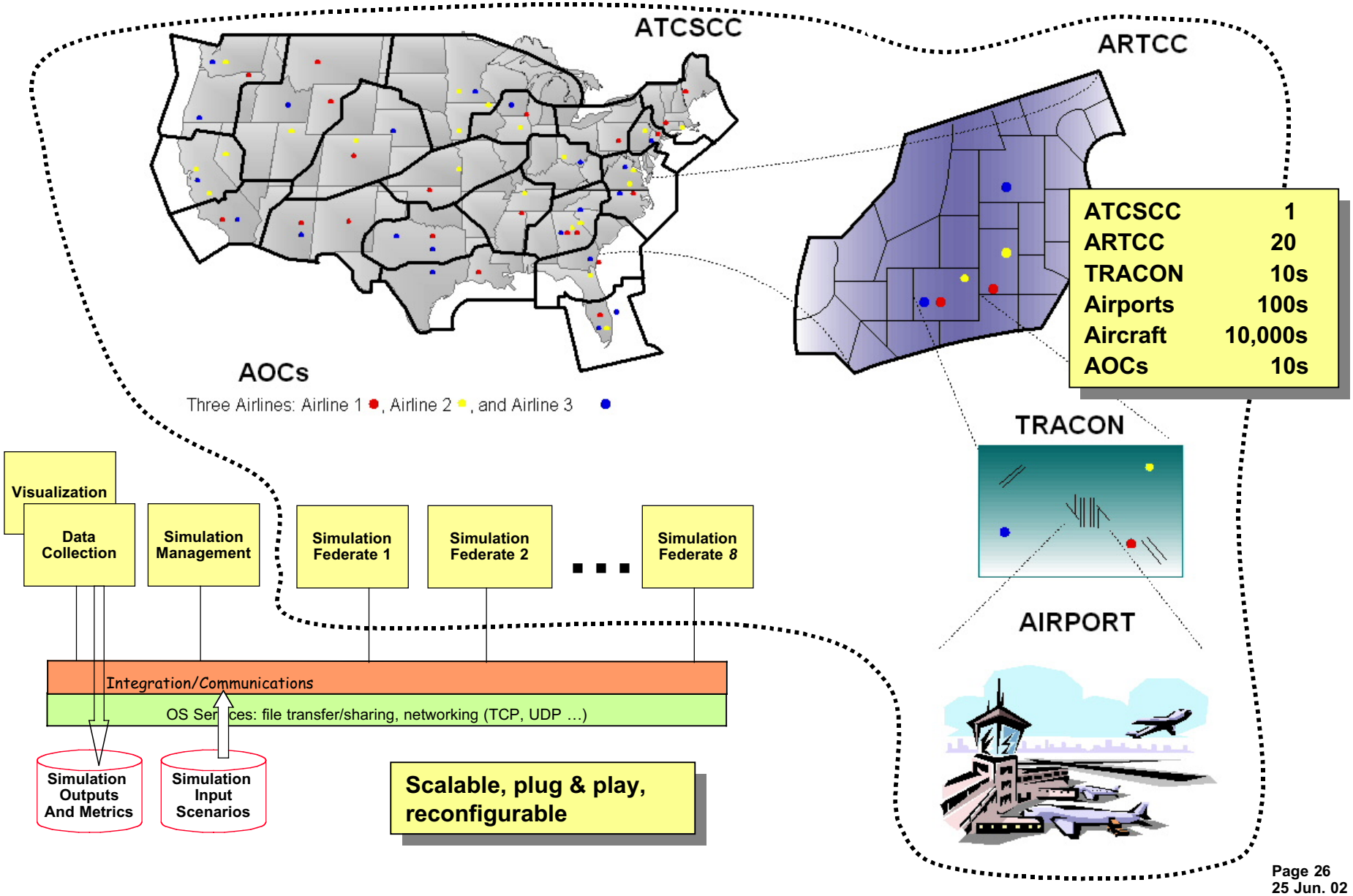
- **Provide the modeling toolbox foundation**
  - Emulate the current NAS operational environment
  - Support NAS-wide, gate-to-gate simulation
  - Ability to model entire day-in-the-NAS scenario
  - Emphasis on modeling Traffic Flow Management interactions
    (including Command Center, ATC, and airlines)
  - En Route ATC (CD&R, speed / vector advisories)
  - Simple terminal and airport models (generic vs specific)
  - Varying degrees of AC model fidelity

ACES Build 1 System

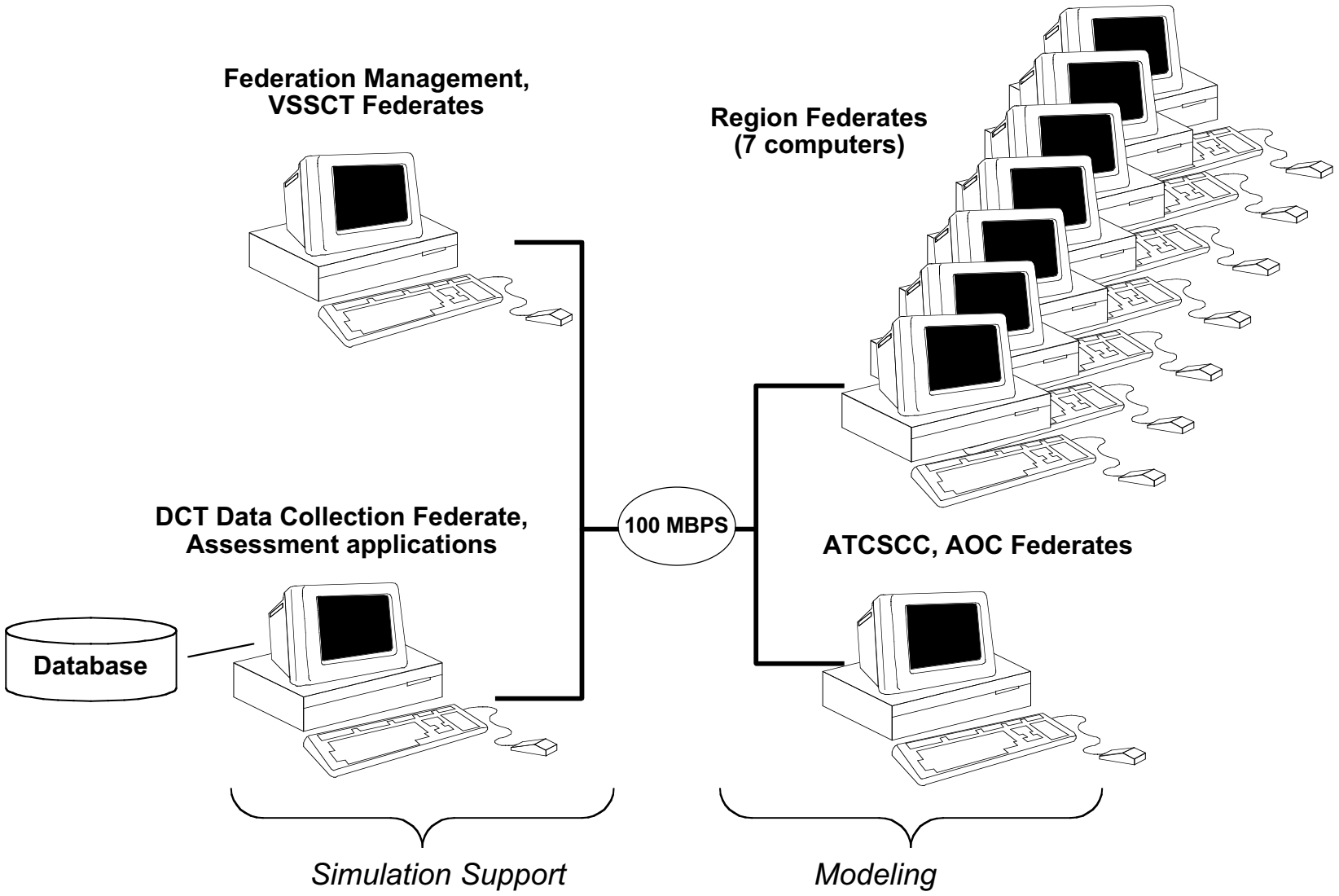
- **Assessment capabilities**
  - Measure delay (gate, taxi, airborne)
  - Fuel costs
  - Controller workload (# of vectors, speed changes, # TFM restrictions, CD&R activity)
  - TFM activity

- **Validate with real world data**

ACES Build 1 System



ACES Build 1 Hardware Configuration



All computers shown are high-end Windows 2000 workstations

Example Assessment Scenarios ACES Build 1 System Could Support

For the current NAS operational environment:

- SCENARIO #1: Assess NAS-wide effects of increasing en-route sector capacities by 25% for a given traffic scenario and a given set of TFM disturbances
- SCENARIO #2: Assess NAS-wide effects of increasing selected airport capacities for a given traffic scenario and a given set of TFM disturbances
- SCENARIO #3: Assess NAS-wide effects of reduced separation standards for a given traffic scenario and a given set of TFM disturbances
- SCENARIO #4: Assess NAS-wide effects of pre and post 911 traffic mix to a given set of TFM disturbances
- SCENARIO #5: Assess NAS-wide effects of planned airport expansions under given set of TFM disturbances and a given traffic demand (current, 2010, 2020?)

Summary

- **ACES integrated architectural and agent-based modeling approach provide:**
  - a flexible, distributed simulation environment
  - a multi-fidelity “modeling toolkit” to support tailored simulations
  - a simulation environment designed for change
- **Prototype system**
  - small scale proof-of-concept version of ACES
  - demonstrated key features of ACES approach
- **Build 1 system**
  - significant increase in scope over proof-of-concept system
  - in development