



Cluster Node Computing for Target Generation Systems in Aircraft Simulations

Spencer Monheim, Michael Feher, James Murphy

NASA Ames Research Center

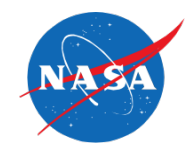
Bonnie Andro-Avila

MIRACORP, Inc.

AIAA SciTech 2019 Conference

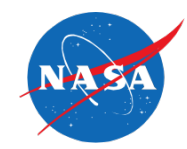
Jan 7-11, 2019

Discuss the development of a general purpose prototype target generator for air traffic management simulations in order to accommodate future research at NASA Ames Research Center's Aviation Systems Division



Outline

- Definition of a Target Generator
- System Design
 - System Design Philosophy
 - Rationale for Language Used
 - System Architecture
 - Rationale for Networking Protocols Used
- Performance Results



Background: What is a Target Generator?

- Simulates multiple aircraft to create an airspace simulation
 - Fly aircraft along pre-defined routes or along vectors
 - Publish the positional and attitude data to an external interface
- Provides interfaces for client software such as pseudopilot or controller interfaces
- Facilitates research simulations



The screenshot displays a flight simulation interface with several panels:

- Target List:** A table listing aircraft parameters.
- Cmd Text Entry:** A text input field for commands.
- State View:** A panel showing aircraft state information.
- Cmd Keys:** A list of function keys and their corresponding actions.
- Notifications 8:** A list of system notifications.
- Main Map View:** A large map showing the airport layout with various aircraft icons and flight paths.

State	Spot	Runway	Callsign	Gate
Hold @...	24	36R	AAL1925	C10
Hold @...	25	36R	AAL1910	C12
Taxi 5...	24	36R	AAL1830	C14
Hold @...	24	36R	AAL2013	C4
Hold @...	24	36R	AAL1832	C6
Airborne	noA1 rport	-1	AAL1770	noA1 rport
Hold @...	24	36R	AAL852	D1
Taxi 7...	24	36R	AAL840	D11
Hold @...	24	36R	AAL1965	D13
Hold @...	24	36R	AAL835	D2
Hold @...	24	36R	AAL1709	D3
Hold @...	24	36C	AAL829	D5
Taxi 7...	24	36R	AAL883	D7
Hold @...	24	36R	AAL1982	D9

State View:

Knots:
State:
Destination:
Gate:
Runway:

Cmd Keys:

- F5 STOP
- F6 GO
- F7 RightOfWay
- F8 Clear_Rte
- F9 Taxi Perimeter2Sp9/Spot9e [HS Spot23]

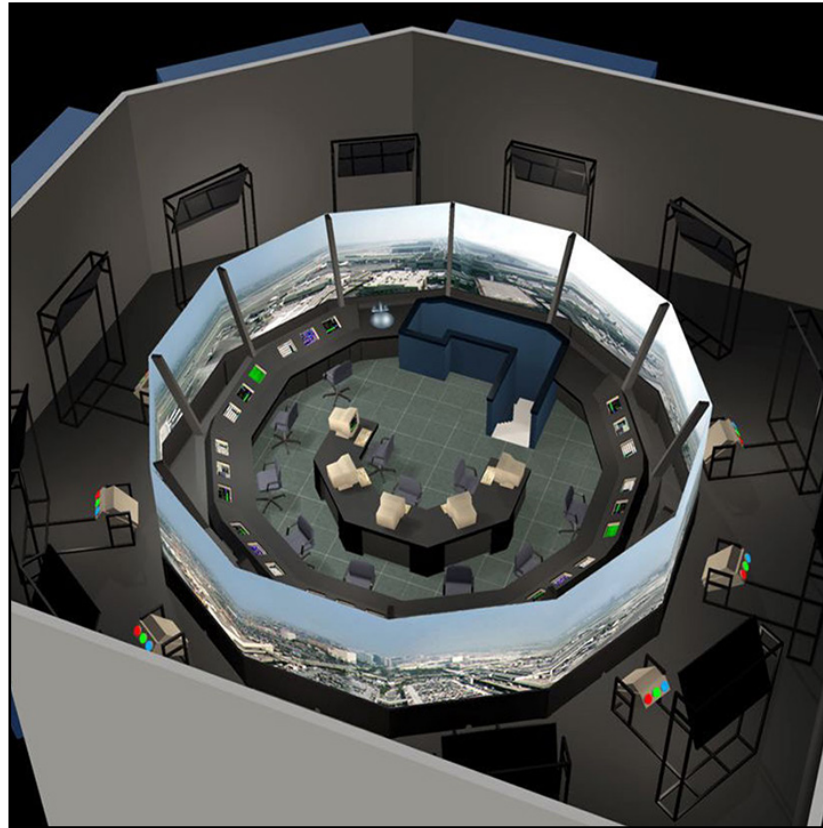
Notifications 8:

- NOTIFY Controller [1] AAL1770 Ready to Push off
- NOTIFY Controller [5] AAL1830 Ready to Push off
- NOTIFY Controller [1] AAL1770 Ready to taxi
- NOTIFY Controller [5] AAL1830 Ready to taxi
- NOTIFY Controller [28] AAL840 Ready to Push off
- NOTIFY Controller [28] AAL840 Ready to taxi
- NOTIFY Controller [26] AAL829 Ready to Push off
- NOTIFY Controller [29] AAL883 Ready to Push off



FutureFlight Central

- Air traffic control simulation facility
- 360 degree visualization
- Facilitates air traffic management research simulations





Building a Target Generator Prototype - Part 1: System Design Philosophy

- Leverage the fact that newer processors are more commonly increasing the number of cores over single-core performance
- Separate the target generator entirely from client software (pilot stations, etc.) and utilize the UNIX-style approach to software
- Provide the capability for pilot and controller interfaces to be remotely accessible via a web browser



Building a Target Generator Prototype - Part 2: Language

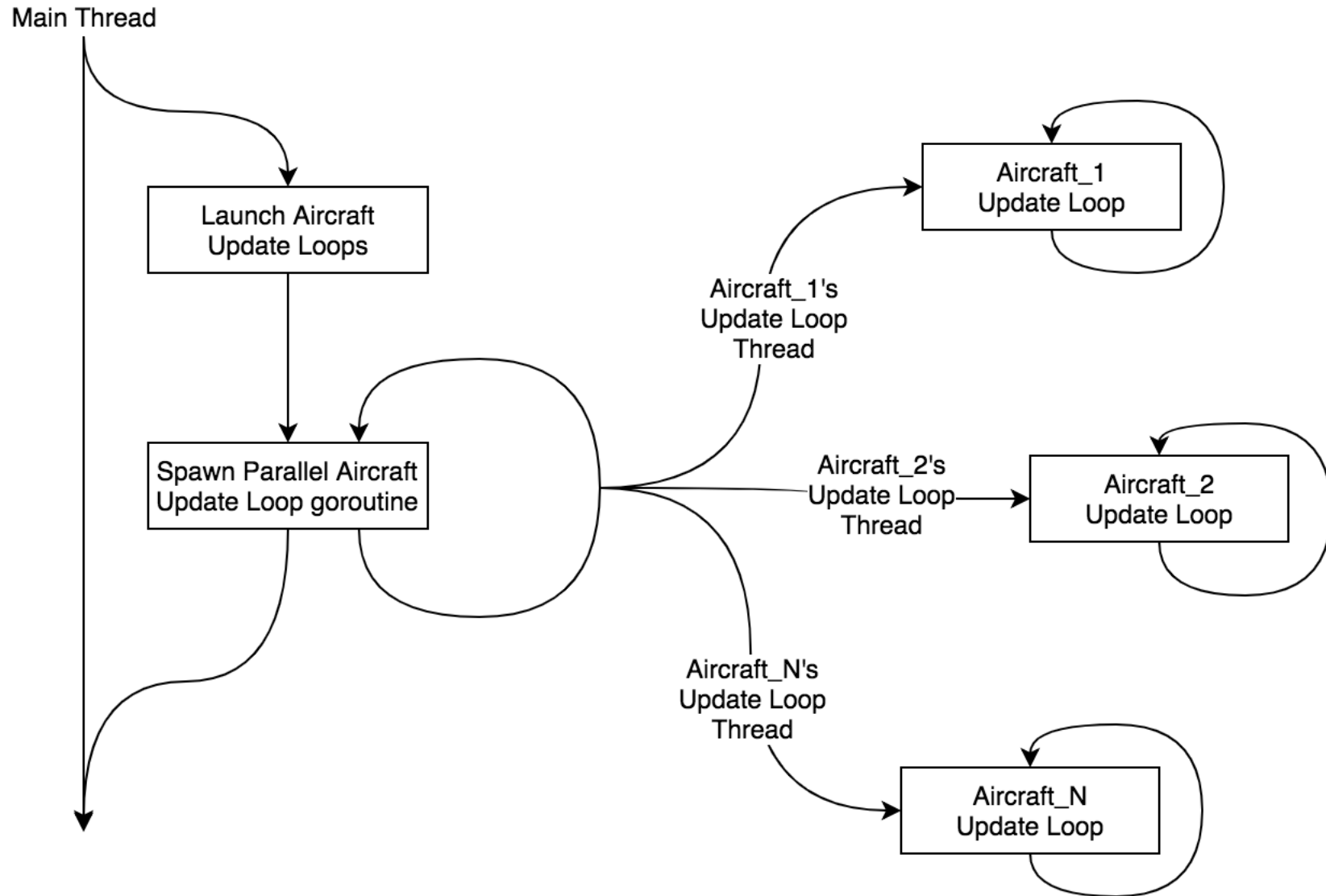
Programming Language Chosen: Go

Selection Criteria:

- Native support for multi-threading
- High developer velocity
- Support multiple network protocols like TCP, UDP, UNIX Domain Sockets, http, Websockets, etc.



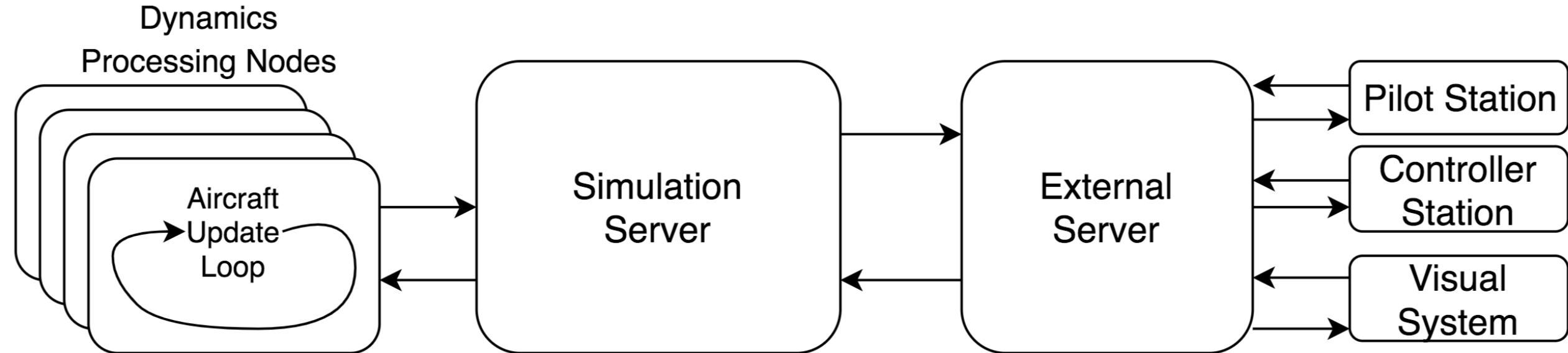
Using Go's concurrency model to implement Parallelism





Building a Target Generator Prototype - Part 3: System Architecture

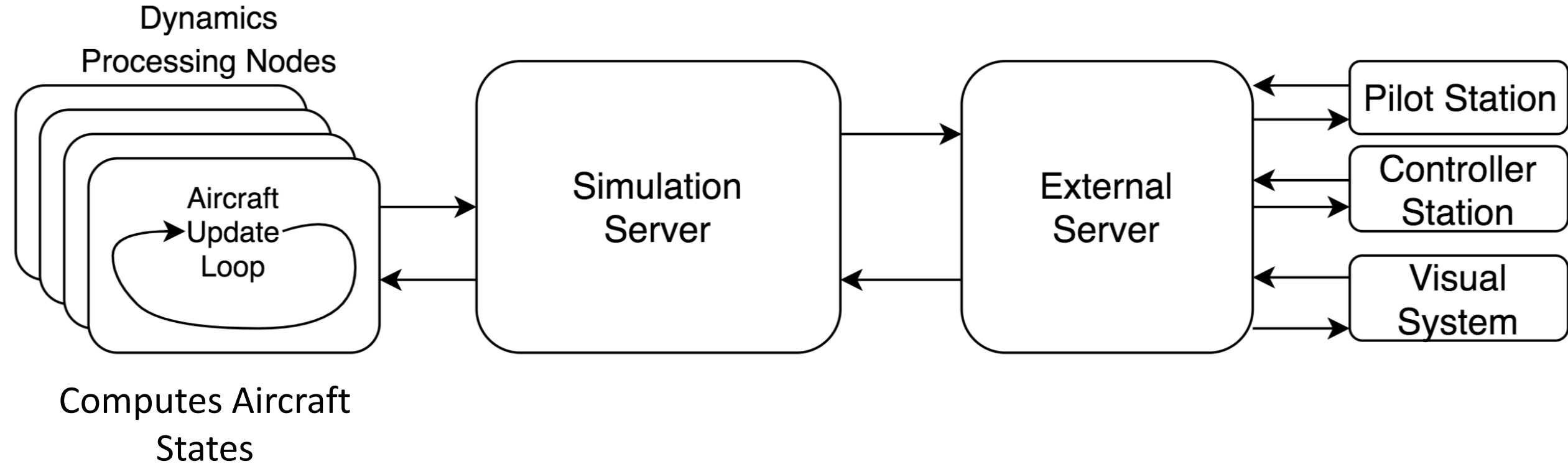
Utilize the UNIX philosophy, “Write programs that do one thing and do it well, write programs that work together.”





Building a Target Generator Prototype - Part 3: System Architecture

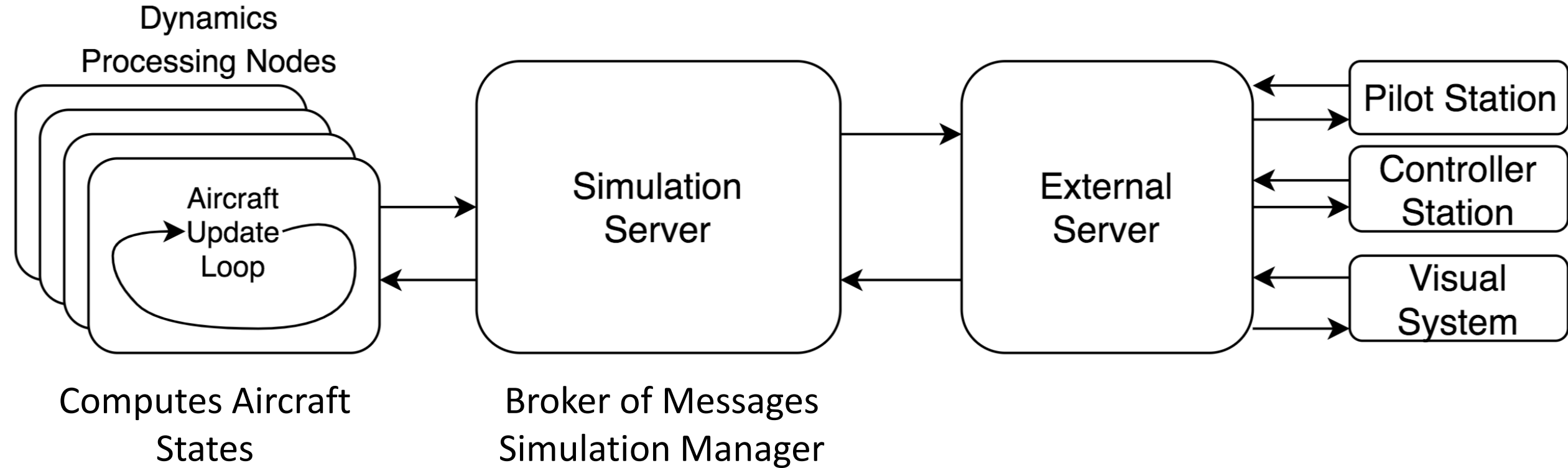
Utilize the UNIX philosophy, “Write programs that do one thing and do it well, write programs that work together.”





Building a Target Generator Prototype - Part 3: System Architecture

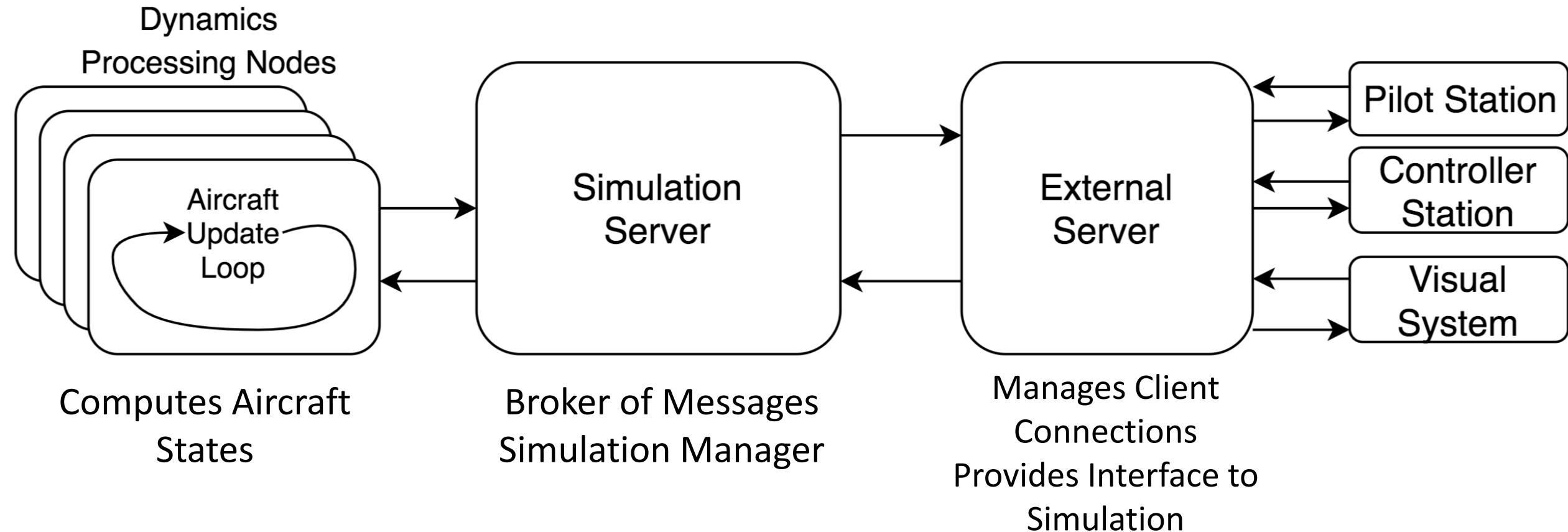
Utilize the UNIX philosophy, “Write programs that do one thing and do it well, write programs that work together.”





Building a Target Generator Prototype - Part 3: System Architecture

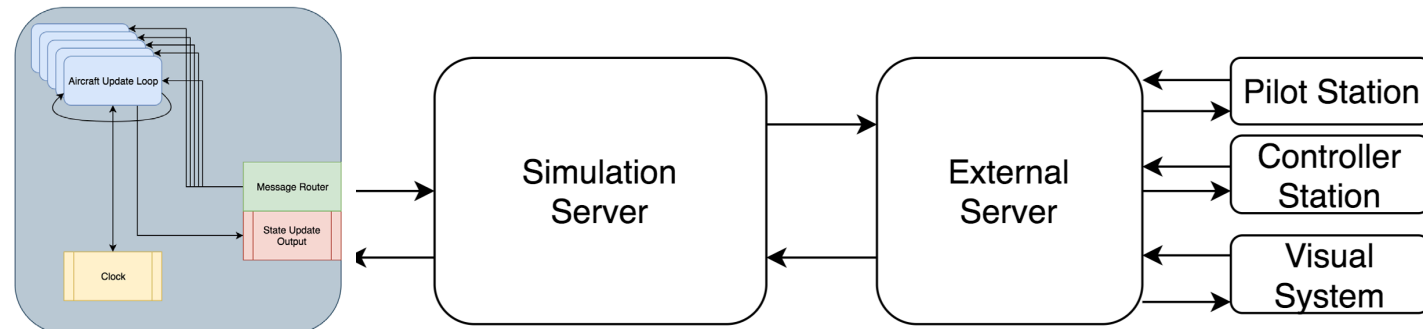
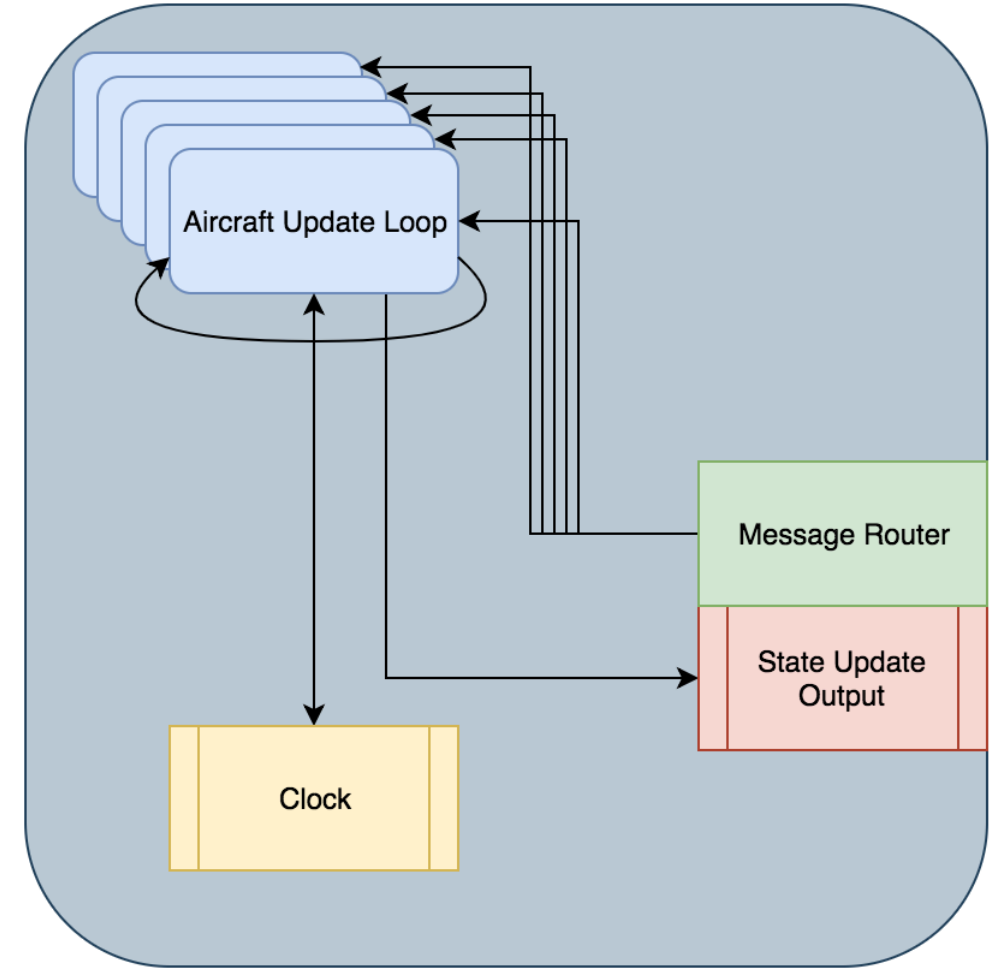
Utilize the UNIX philosophy, “Write programs that do one thing and do it well, write programs that work together.”





Dynamics Processing Node

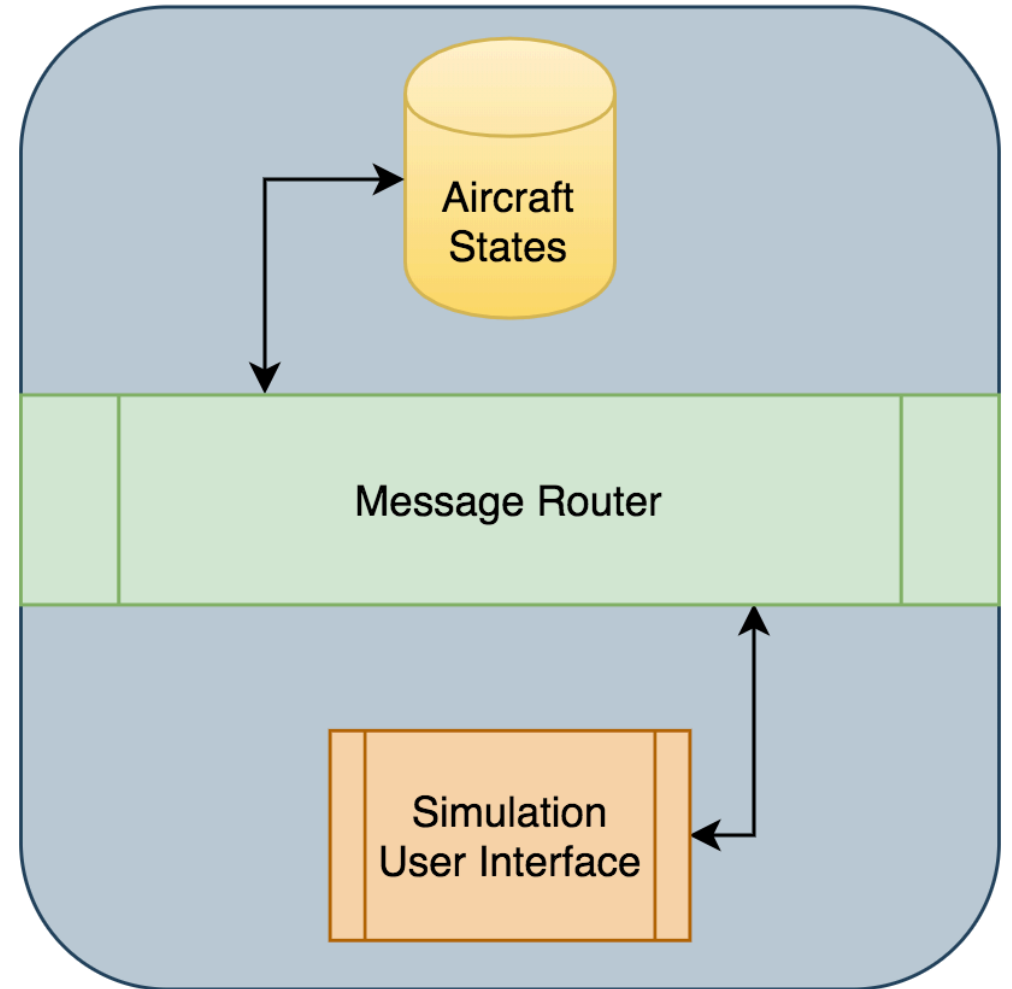
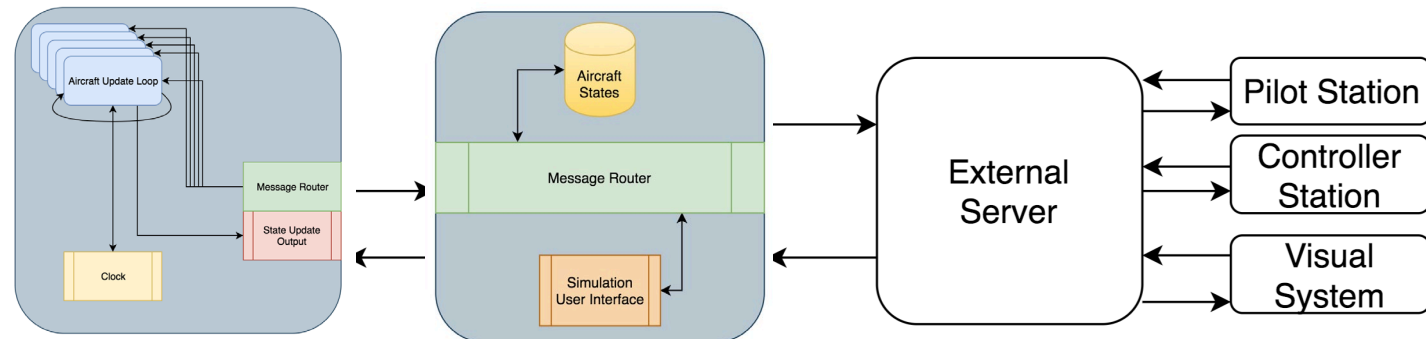
- Clock
- Aircraft Update Loop
- Message Router
- State Update Output





Simulation Server

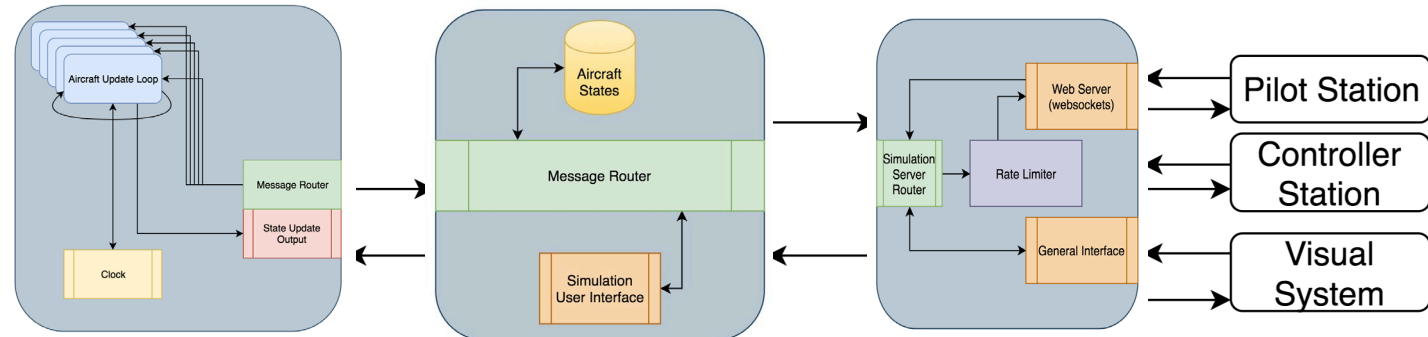
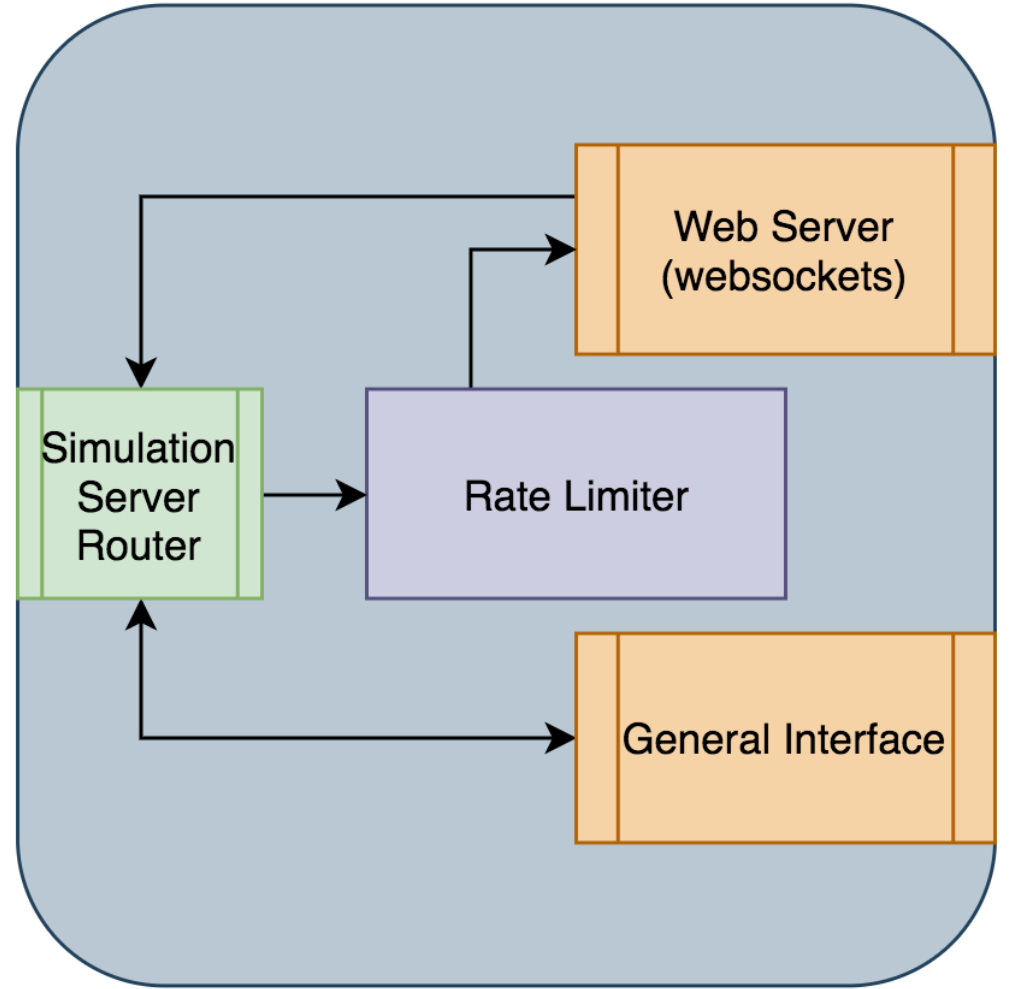
- Message Router
- Aircraft States
- Simulation User Interface





External Server

- Simulation Server Router
- Rate Limiter
- Web Server
- Web Server
- General Interface





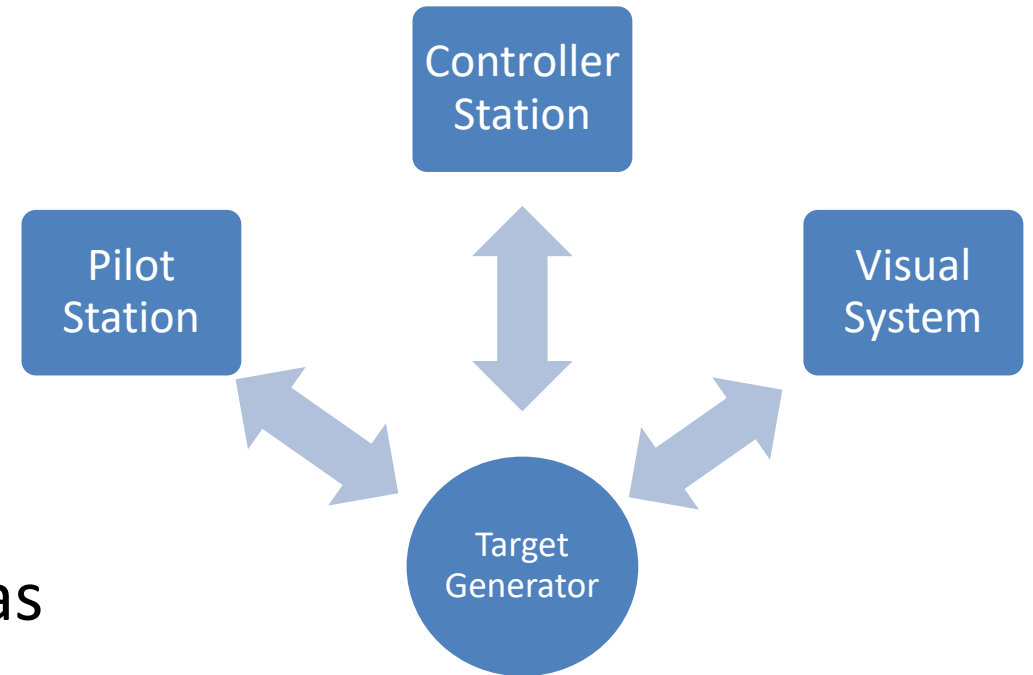
Building a Target Generator Prototype - Part 3: Network Protocols

	Internal Simulation Communication	Low Bandwidth Clients	High Bandwidth Clients
Local Communication	UNIX Sockets	N/A (Only Distributed)	N/A (Only Distributed)
Distributed Communication	TCP	HTTP/Websockets	TCP



Why WebSockets and HTTP for Pilot and Controller Stations?

- Browser Web Applications allow for ease-of-access from any on-site location or VPN'd client
- WebSockets provide streaming data to support a real-time connection to simulation
- HTTP enables upgrading to websockets, as well as a strong interface for request-response communication





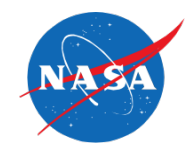
Results

- Current day target generators offer 1 - 4Hz, up to 200 Active Aircraft

Prototype Tests:

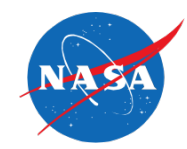
- Two system configurations
 - one all-local simulation
 - one distributed system
- Two loads of traffic
 - one 400 aircraft
 - one 1000 aircraft

	400 Aircraft	1000 Aircraft
One Node	125.4Hz	48.9Hz
Two Node	79.7Hz	32.0Hz



Next Steps

- One Node Configurations outperformed Two Node Configurations:
 - Need to optimize messaging system for TCP
- Develop Clients
 - Pilot Station
 - Controller Station
- Interface with Visual Systems



Summary

- Prototype demonstrates update rates of 30Hz+ outperforming current day target generators
- Provides smooth out-the-window visuals
- Supports higher aircraft density
- Expands flexibility for client software
- Improves capabilities for future simulations research