Project Report
ATC-361

NextGen Weather Processor Architecture Study

O.J.Newell
M.M. Wolfson
E.R. Ducot

TBD February 2010

Lincoln Laboratory
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LEXINGTON, MASSACHUSETTS

Prepared for the Federal Aviation Administration,
Washington, D.C. 20591

TECHNICAL REPORT STANDARD TITLE PAGE

1. Report No. ATC-361	2. Government Accession No.	3. Recipient's Catalog No.

4. Title and Subtitle NextGen Weather Processor Architecture Study	5. Report Date TBD Februaty 2010 6. Performing Organization Code
7. Author(s) O.J. Newell, M.M. Wolfson, and E.R. Ducot	8. Performing Organization Report No. ATC-361
9. Performing Organization Name and Address MIT Lincoln Laboratory 244 Wood Street Lexington, MA 02420-9108	10. Work Unit No. (TRAIS) 11. Contract or Grant No. FA8721-05-C-0002
12. Sponsoring Agency Name and Address Department of Transportation Federal Aviation Administration 800 Independence Ave., S.W. Washington, DC 20591	13. Type of Report and Period Covered Project Report 14. Sponsoring Agency Code

15. Supplementary Notes

This report is based on studies performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology, under Air Force Contract FA8721-05-C-0002.

16. Abstract

The long-term objectives for the NextGen Weather Processor (NWP) include consolidation of today's multiple weather systems, incorporation of recent and emerging Federal Aviation Administration (FAA) infrastructure (Federal Telecommunications Infrastructure (FTI), System Wide Information Management (SWIM), NextGen Network-Enabled Weather (NNEW)), leveraging National Oceanic and Atmospheric Administration (NOAA) and/or commercial weather resources, and providing a solid development and run-time platform for advanced aviation weather capabilities. These objectives are to be achieved in a staged fashion, ideally with new components coming on-line in time to replace existing capabilities prior to their end-of-life dates.

As part of NWP Segment 1, a number of alternative implementations for the NWP as it might exist in the 2013 time frame have been proposed. This report examines the alternatives from a top-down technical perspective, assessing how well each maps to a high-level NWP architecture consistent with the long-term NextGen information sharing vision. Technical challenges and opportunities for weather product improvements associated with each alternative are discussed. Additional alternatives consistent with the high-level NWP architecture, as well as a number of suggested follow-on analysis efforts are also presented.

17. Key Words	18. Distribution Statement This document is available to the public through the National Technical Information Service, Springfield, VA 22161.

19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 65	22. Price

FORM DOT F 1700.7 (8-72) Reproduction of completed page authorized

ABSTRACT

The long-term objectives for the NextGen Weather Processor (NWP) include consolidation of today’s multiple weather systems, incorporation of recent and emerging Federal Aviation Administration (FAA) infrastructure (Federal Telecommunications Infrastructure (FTI), System Wide Information Management (SWIM), NextGen Network-Enabled Weather (NNEW)), leveraging National Oceanic and Atmospheric Administration (NOAA) and/or commercial weather resources, and providing a solid development and run-time platform for advanced aviation weather capabilities. These objectives are to be achieved in a staged fashion, ideally with new components coming on-line in time to replace existing capabilities prior to their end-of-life dates.

As part of NWP Segment 1, a number of alternative implementations for the NWP as it might exist in the 2013 time frame have been proposed. This report examines the alternatives from a top-down technical perspective, assessing how well each maps to a high-level NWP architecture consistent with the long-term NextGen information sharing vision. Technical challenges and opportunities for weather product improvements associated with each alternative are discussed. Additional alternatives consistent with the high-level NWP architecture, as well as a number of suggested follow-on analysis efforts are also presented.

TABLE OF CONTENTS

Page

TABLE OF CONTENTS (CONTINUED)

Page

TABLE OF CONTENTS (CONTINUED)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Continued)

LIST OF TABLES

This page intentionally left blank.

1. INTRODUCTION

1.1 SCOPE

The long-term objectives for the NextGen Weather Processor (NWP) include consolidation of today's multiple weather systems, incorporation of recent and emerging FAA infrastructure (Federal Telecommunications Infrastructure (FTI), System Wide Information Management (SWIM), NextGen Network-Enabled Weather (NNEW)), leveraging National Oceanic and Atmospheric Administration (NOAA) and/or commercial weather resources, and providing a solid development and run-time platform for advanced aviation weather capabilities. These objectives are to be achieved in a staged fashion, ideally with new components coming on-line in time to replace existing capabilities prior to their end-of-life dates.

As part of NWP Segment 1, three main alternative implementations for the NWP as it might exist in the 2013 time frame have been proposed. Each of the three alternatives has multiple subalternatives, bringing the total number of alternatives to eight. The alternatives are currently defined at a very high level. Additional details are required to provide the necessary information to both refine and score the alternatives against one another. This report examines the alternatives from a top-down technical perspective, assessing how well each maps to a high-level NWP architecture consistent with the long-term NextGen information sharing vision. Technical challenges and opportunities for product improvements associated with each path are discussed. Specific costs associated with the alternatives are beyond the scope of this document, and are discussed only in a qualitative sense.

1.2 BACKGROUND

1.2.1 Definition of Terms

The term "NextGen Weather Processor" naturally evokes an image of a centralized processor, perhaps with an associated backup system that physically resides at a separate location. In the context of NextGen, however, the term actually refers to the aggregate processing capability required to generate the weather products of interest to the aviation weather community, whether implemented in a centralized or more distributed fashion. Distributed implementations can encompass not only geographically diverse processing elements, but organizationally diverse processing elements as well (e.g., FAA, commercial vendor).

It is useful to define what is meant by the term *NWP Architecture* in the context of this report. The term *architecture* is very broad, and architectures can exist at many levels of detail and address multiple viewpoints. *System architecture* and *information architecture* are commonly used specializations of the term that address higher-level architectural issues such as network topologies and globally addressable and accessible data items. At a lower-level of detail, the term *computer architecture* is often used to refer to physical processor configurations and interconnect topologies.

The term *architecture* itself warrants further discussion, as it is often used interchangeably with *design*, especially within the Information Technology (IT) community. Though the line between the two terms is fuzzy, it is useful to maintain a distinction between them. In this study, *architecture* refers to the overall framework within which physical instances of a system can be constructed – the set of long-term high-level guiding principles. The specific *design* of a system at a certain point in time can be said to *conform* to the overall architecture, rather than itself constituting an architecture. To borrow an analogy from the housing industry, there can be many house *designs* that conform to the Victorian *architecture*. To be consistent with this usage, the NWP alternatives discussed later in this report are referred to as alternative *designs* that are all consistent with the higher-level NWP architecture, rather than "NWP architecture alternatives."

1.2.2 NextGen Weather System Transformation

A variety of weather systems are in use today in the National Airspace System (NAS), ranging from local terminal solutions, such as the Terminal Doppler Weather Radar (TDWR) and Airport Surveillance Radar-9 (ASR-9), to regional solutions such as Integrated Terminal Weather System (ITWS), and finally to continental United States (CONUS)–wide solutions, such as Weather and Radar Processor (WARP) and Corridor Integrated Weather System (CIWS). Though the regional and CONUS-scale systems do ingest data from the local terminal sensors (TDWR, ASR-9), each system is largely self-contained and is built using system-specific infrastructure components to address input/output (I/O), processing, and display functionality. Figure 1 depicts three of the core weather systems and some of the high-level components associated with each. In a net-centric environment, such systems are said to be "stove-piped" since they are not typically designed to enable re-use of common software and hardware components, and sharing of information between systems is difficult.

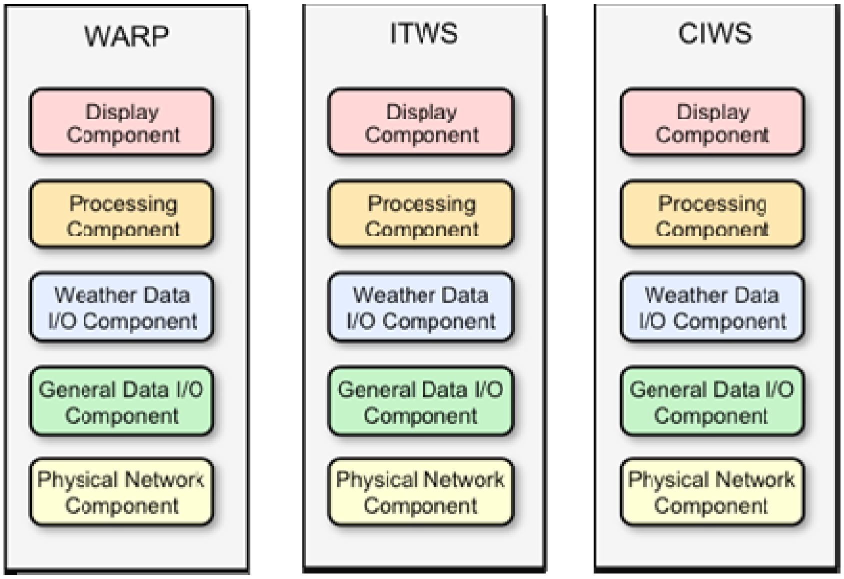


Figure 1. Examples of stove-piped FAA weather systems.

Difficulties associated with the current stove-piped weather system model include:

1. ***Complex system software and hardware maintenance.*** Software components are not shared to the extent they could be, increasing initial and maintenance costs. Likewise, hardware components, though many are commercial off-the-shelf (COTS) based, are unnecessarily diverse, increasing hardware maintenance costs.

2. ***Lack of common situational awareness.*** Multiple systems tend to produce different depictions of weather events, depending on the types of input sensors and the data processing involved. The WARP national precipitation mosaic, for example, differs in subtle yet significant ways from the CIWS national mosaic. Even for the region of the country covered by an individual ITWS's long-range mosaic, this product differs from either view provided by WARP or CIWS covering the same space. Common situational awareness is one of the core issues to be addressed in NextGen.

3. ***Lack of ability to easily share data among systems.*** The trend over time is to supplant single-sensor weather products with products generated by data fusion algorithms fed by multiple sensors, for reasons of data quality as well as to provide the common situational awareness mentioned previously. This trend is enabled by the increasing availability of communications bandwidth, NAS-wide.

4. ***Multiple weather displays.*** The existence of multiple weather displays, especially in the crowded tower cab environment, is detrimental from an end-user controller perspective as well as a maintenance perspective.

5. ***Non-scalable processing.*** Processors associated with these systems tend to scale to a certain extent, but no further. This makes them inherently difficult to adapt to the NextGen data fusion environment.

In NextGen, the previously stove-piped weather system model is being transformed to a model organized around layered capabilities. This is a fundamental shift in the system architecture, in some sense corresponding to a 90 rotation of the weather system "architectural axis." The transformation is depicted in Figure 2. As shown in the diagram, this shift essentially requires a similar transformation in the organizational structure of weather programs to align with the layered architecture[1], a shift that is already underway. In essence, many of the components embedded within today's vertically aligned weather programs will be replaced by common functionality in the horizontally oriented programs shown. The assumption of this report is that the processing components for all weather programs are to be

---

[1] Based on corollary to Conway's Law, which states *"Any organization that designs a system will necessarily produce a design whose structure is a copy of the organization's communication structure."*

consolidated under a new processing-focused program, labeled "NWP" in the diagram. Note that the eventual home for the existing weather-specific display components is shown as TBD, as the future requirements for such displays have not yet been determined.

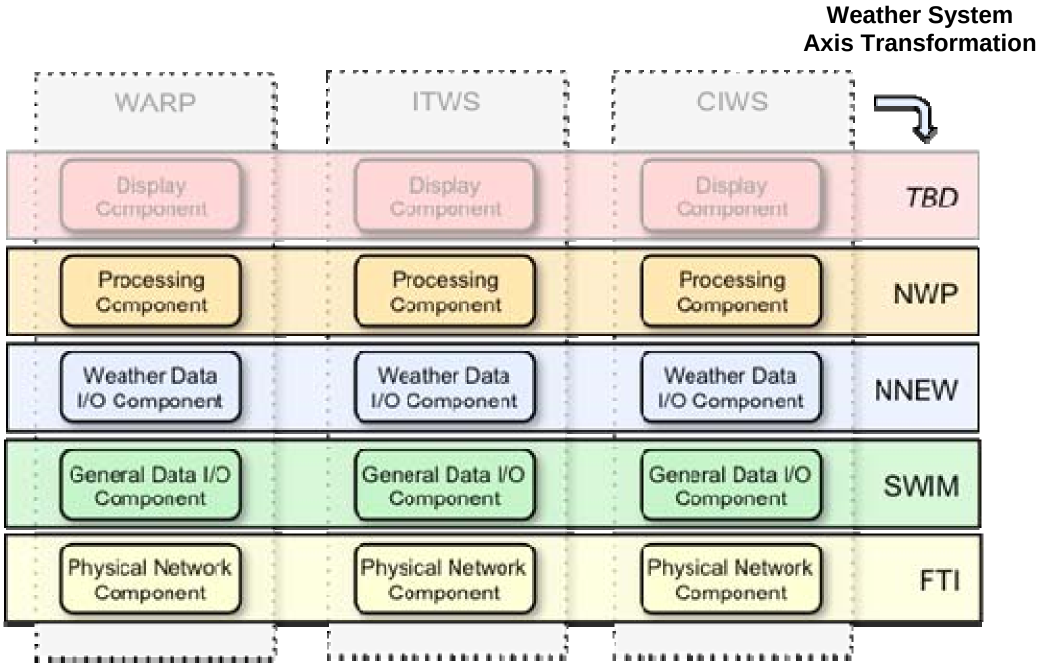


Figure 2. The same weather capabilities as shown in Figure 1, reorganized around the NextGen layered capability model.

Given this transformation, what are the important focus areas for an NWP architecture? From the *system* and *information* architecture perspective, we want to ensure that the capabilities in the other architectural layers are leveraged and the interfaces between the processing layer and those layers are clearly defined and architecturally consistent. We also require that the system architecture supports a variety of deployment topologies, as it is clear that the NWP topology will be changing over time due to the staged development approach. From a *computer* architecture perspective, we want the NWP compute platform to be flexible enough to address a variety of processing needs, while remaining cost-effective.

1.3 PROPOSED SEGMENT 1 NWP IMPLEMENTATION ALTERNATIVES

To encourage system design solutions that are both innovative and cost-effective, the FAA system development process requires that multiple implementation alternatives be developed and explored. The challenge with respect to the segment 1 NWP design is to identify the dimensions along which implementation flexibility exists. From the systems and information architecture perspective, the NWP design is constrained, and in fact driven by, the architecture embodied in the FTI, SWIM, and NNEW layers. From a computer architecture perspective, processing has increasingly become a commodity item, even in the context of large weather model computation tasks.

The primary dimension along which NWP implementation flexibility exists is system topology, both in the geographical and organizational sense. Which processing components of the NWP are distributed and which are centralized? Which agency, organization, or commercial entity is responsible for which set of weather data products in the long term? More secondary considerations include how to best leverage existing systems and how to best stage the overall development to minimize throwaway efforts. These questions in fact form much of the basis for the current set of NWP alternatives.

The list of top-level alternatives for NWP Segment 1 is shown in Table 1. The first main alternative represents an NWP implementation that resides entirely within the FAA domain. The second and third alternatives represent NWP implementations that reside in part or entirely within the NOAA or commercial domains, respectively. Multiple subalternatives exist within each main category and are described in more detail in Section 6.

TABLE 1

Proposed NWP Segment 1 Alternatives

FAA	FAA produces advanced forecast products and legacy products. FAA publishes and subscribes to products
NOAA	NOAA provides advanced forecast products and optionally FAA legacy products
Commercial Vendor	Commercial vendor provides advanced forecast products and optionally FAA legacy products

All of the alternatives described in the table above are technically feasible, though some are certainly more challenging and provide more or less potential near-term benefit than others. Alternatives that include NOAA or a commercial vendor depend on a number of external factors that are beyond the scope of this study, though an attempt is made to identify high-risk areas where possible.

1.4 REPORT ORGANIZATION

The remainder of this report is organized as follows. Section 2 provides a brief overview of the FTI, SWIM, and NNEW infrastructure programs, with a focus on the capabilities that tend to influence the overall NWP architecture. Section 3 identifies some of the key computing trends relevant to the NWP. Section 4 describes a notional NWP architecture at a high level. Section 5 describes the key FAA systems involved in the alternatives, assessing how well each maps to the high-level NWP architecture. Section 6 defines a framework within which to qualitatively score how well matched the different alternatives map to the NWP architecture and scores each alternative with respect to those metrics. Section 7 summarizes the results and provides recommendations for additional follow-on work.

2. NEXTGEN INFRASTRUCTURE PROGRAMS

2.1 OVERVIEW

One of the key objectives for NWP Segment 1 is to align with and/or leverage the emerging NextGen infrastructure where possible. This section provides an overview of three key infrastructure programs, FTI, SWIM, and NNEW, and identifies areas that potentially impact the NWP architecture. As described in Section 1.2.2, FTI, SWIM, and NNEW provide NWP with an infrastructure stack that includes physical network, general information management, and weather-specific information management layers. These are discussed starting with the lowest layer and moving up in the stack.

2.2 FAA TELECOMMUNICATIONS INFRASTRUCTURE

2.2.1 Capabilities

The FTI provides Internet Protocol (IP)–based network communications comprising redundant and fault-tolerant network backbone (Figure 3). As shown in the figure, the backbone, currently built atop leased lines from both Sprint and AT&T, connects different users located at Air Route Traffic Control Center (ARTCC) and Terminal Control Center (TRACONS). These user locations, referred to as *nodes* in the figure, are either connected to FTI using a fully meshed topology (shown in yellow) or connected to at least two fully meshed nodes (shown in green). The intent of the design is to support the high levels of reliability required by both surveillance data and critical weather data.

Connections to FTI provide a fixed guaranteed bandwidth, where the bandwidth reflects the requirements of each connection. The end-to-end latency between any two points in the network is bounded at 250 ms, a requirement largely driven by the needs of the surveillance community. FTI is currently in the process of transitioning to an optical backbone, which is expected to increase the available bandwidth in the core network by at least an order of magnitude. The maximum end-to-end packet latency between any two points in the network is expected to decrease significantly as well (<100 ms).

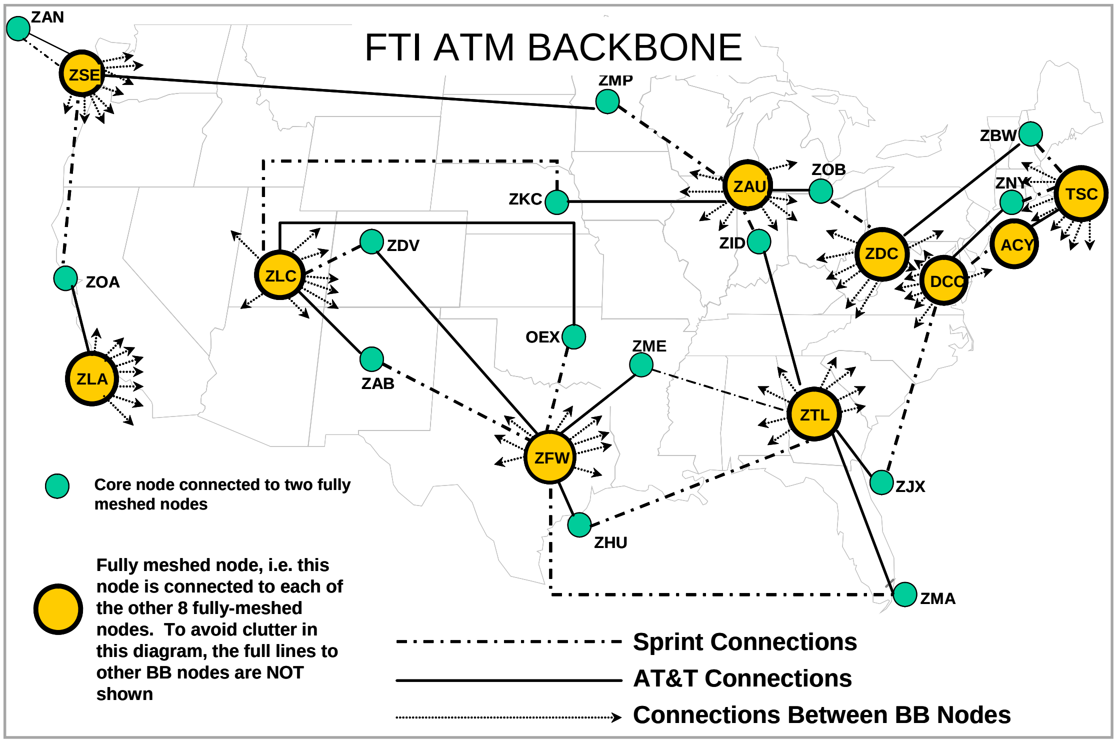


Figure 3. FTI network backbone.

2.2.2 Gaps

FTI provides high-reliability connections, but does not currently provide a means for differentiated message traffic based on priority over those connections. At the physical network level, there is currently no means to distinguish, for example, high-priority wind-shear message traffic from lower-priority (but large) weather reflectivity data. This is problematic since traffic from low priority users can overflow the queues of the edge routers and introduce unacceptable latency for higher-priority traffic. If a single FTI connection is to be used between a weather data provider and a weather data consumer, traffic classification and prioritization must be handled solely at the application level. This is a suboptimal solution in terms of software complexity and efficient use of network bandwidth. The alternative, using separate links for the different traffic classes, is possible but not a good long-term solution from a cost perspective.

The FTI program is currently evaluating the possibility of enabling traffic prioritization in the core FTI network. To some extent it is up to the FTI user community, and in particular the high-volume weather data community, to help drive the requirement. Investigation into the more detailed requirements and possible solutions that span not only the FTI layer, but the SWIM and NNEW layers as well, is currently being conducted in the context of the NNEW program, A second potential gap exists at the boundary between FTI and other external networks. Government-wide policy dictates that the number of connections between government organizations and external networks be consolidated, for manageability and security reasons. The overall goal is to reduce the number of interconnection nodes to less than 70. The FAA has currently been allocated 4 of these network nodes, and the FTI program is in the process of establishing gateways at selected locations. Preliminary measurements of the throughput supported by these gateways indicate relatively low overall throughput when compared with the bandwidths required for passing of large gridded weather data sets in a timely fashion. Both the FTI and the SWIM programs are currently examining this issue. Again, it is up to the weather providers and consumers to help drive the overall bandwidth requirement.

2.2.3 FTI Impact on NWP Architecture

Availability of cost-effective, reliable, high-bandwidth communications within the NAS and between the NAS and government and commercial vendors is a significant long-term driver for the NWP system architecture. Expensive and/or low-reliability communications links drive a processing architecture to a more distributed model, where processing relevant to a given region of interest is performed locally. As the cost of the communications decreases and its reliability increases, a more centralized model becomes attractive for maintenance, scalability, and data fusion opportunity reasons.

Latency budgets for weather systems are typically measured in seconds rather than milliseconds. One of the more stringent latency requirements for weather data, for example, is the requirement for wind shear data (ITWS, TDWR, ASR-Weather Systems Processor (WSP)), currently specified as on the order of 10 seconds from the time of detection to the time it is displayed to controllers and relayed to pilots. This is well within the latency bounds provided by FTI on a single link. A similar situation exists for ASR-9 weather channel data – there is a 5-second window in which to transmit the final product to the controller displays following generation of a 30-second update. Again, this is not a particularly demanding requirement for modern communications links.

For a single product travelling on a single communications line, FTI as it exists today is capable of meeting weather traffic latency requirements. When multiple products that compete for available bandwidth on a single connection are considered, however, the lack of traffic classification in FTI becomes an obstacle to centralization of existing capabilities. If this shortcoming is not addressed in the segment 1 time frame, the NWP must continue to support a distributed architecture with local processing of high-priority, low-latency products such as wind shear. If traffic classification is implemented by the FTI vendor in the NWP Segment 2 time frame, the local processing constraint will largely disappear, opening up the centralized processing option to even high-priority weather data.

With respect to the architectural impact of FTI Gateway throughput, we believe that this is a relatively minor issue than should be addressed in the NWP Segment 1 time frame. The approximate requirements for inter-organizational data transfers, however, need to be specified in the relatively near term to enable the FTI Gateway to be appropriately sized.

2.3 SYSTEM-WIDE INFORMATION MANAGEMENT

2.3.1 Capabilities

The SWIM infrastructure and the NNEW infrastructure described in the next section are both based around the concept of a service-oriented architecture (SOA). Like the term architecture itself, SOA is a broad term that tends to be used in multiple contexts. From the NextGen perspective, perhaps the single most important architectural constraint imposed by an SOA is that functionality be modular and composable, particularly at wide-area network (WAN) scales. From the NWP perspective, this architectural approach lends itself well to either distributed or centralized processing solutions, as well as supporting combinations of the two.

SWIM is intended to provide common standards services such as messaging, monitoring, and security to all NextGen participants. In SWIM Segment 1, this functionality is primarily supplied via the commercially supported open-source FUSE software suite from Progress Software, with additional functionality supplied by, and shared among, SWIM-Implementing Programs (SIPs). The Progress FUSE software suite is based on a set of open source products from the Apache foundation. A number of the key products are listed below.

- ***FUSE Message Broker.*** Based on the Apache ActiveMQ messaging project, this product provides a message broker backbone compliant with the Java Message Service (JMS) specification. It provides a number of built-in features for enabling time-sensitive and reliable message transport. It also provides monitoring hooks based on the Java Management Extension (JMX) to support monitoring of message traffic at the low level.
- ***FUSE Mediation Router.*** Based on the Apache Camel project, this product consists of an extensible core message passing framework and a set of pre-built components that can be used within that framework to implement a wide variety of enterprise integration patterns. In the NWP context, this product is useful for implementing fault-tolerant and load balanced data dissemination solutions, as well as providing an abstraction layer for messaging to support not only JMS, but other transport protocols such as Hypertext Transfer Protocol (HTTP) and Extensible Messaging and Presence Protocol (XMPP).
- ***FUSE Enterprise Service Bus (FUSE ESB).*** Based on Apache ServiceMix, this product provides the runtime environment for Java-based service implementations. FUSE ESB supports a modular, dynamic approach to service deployment, based on the Open Systems Gateway interconnect (OSGi) specification. This technology is theoretically capable of supporting hot-swap software upgrades, though the extent to which that is practical and/or necessary is yet to be determined. FUSE ESB also supports clustering of multiple instances for load-balancing purposes, a feature that may become important for NWP if it is deployed using a more centralized topology.

- ***FUSE HQ.*** Based on the Hyperic HQ product, this product provides a monitoring infrastructure capable of monitoring a wide variety of system conditions out of the box. It is also extensible, easily accommodating additional monitoring inputs that comply with one of the supported monitoring protocols (e.g., JMX)

  From an interoperability perspective, it is important to note that these products are well aligned with SOA standards such as Simple Object Access Protocol (SOAP), Web Services Description Language (WSDL), Extensible Markup Language (XML), and HTTP. Use of these common products within the FAA is encouraged rather than mandated to encourage the use of a common code base. In the case of an NWP implementation that includes other organizations, interoperability should not be compromised if other SOAP product suites are used. In other words, standards compliance is the key, rather than product compliance.

2.3.2 Gaps

The JMS standard, though widely used in SOA applications, does not yet support an “on-the-wire” standard – it is standardized at the application programming interface (API) level only. This issue will likely be resolved in the future if the user community demands an on-the-wire standard, but in the meantime adapters will need to be used if a mix of JMS implementations is used.

Although the FUSE Message Broker software provides a level of support for traffic classification, it is not yet clear if the implementation will be sufficiently robust to meet the NWP message latency requirements. This is a potential gap that will require follow-on evaluation to confirm, as will the ability of the product to propagate traffic classification information to the physical network layer.

Monitoring technologies associated with the FUSE product line are targeted towards SOA infrastructure rather than large-scale processing infrastructure. Monitoring technologies associated with processing clusters (e.g., Ganglia) will need to be bridged to the FUSE monitoring technologies if SWIM is to effectively monitor the NWP processing components. If there is no plan for SWIM to monitor NWP resources, then this is not an issue.

2.3.3 SWIM Impact on NWP Architecture

SWIM is based on the SOA concept, and this in turn influences the NWP architecture. SOA encourages a composable design, allowing for flexibility in terms of the partitioning of distributed and centralized components. For portions of the NWP that reside within the FAA domain, SWIM’s selection of the FUSE product line certainly encourages a Java-based implementation at the interface boundaries. Portions of the NWP that reside at other organizations are free to use an implementation of their choosing, as long as the implementation is compliant with the common SOA standards.

2.4 NEXTGEN NETWORK-ENABLED WEATHER

2.4.1 Capabilities

The NNEW program specifies a set of data standards and data access service standards to be used by all weather providers and consumers. Observational data and data from subsequent fusion systems and forecast models are made available using NNEW standards, forming a 4-D cube of data in dimensions of space and time. It is one of the key enabling programs in the NextGen weather capability portfolio.

The current vision of NNEW entails a hub and spoke architecture comprising origin servers combined with a set of distributed server nodes that intelligently adapt to data access requirements and minimize overall bandwidth demands. The distribution nodes support “fan-in” (e.g., aggregating data) and/or “fan-out” (e.g., splitting data) to make efficient use of the underlying physical network. As clients in the network make requests for data, the cube infrastructure dynamically adapts, requesting data from the origin server and providing it to multiple consumers within the NAS.

Like SWIM, NNEW is based around the concept of SOA, and data formats and data access services are designed using a variety of composable building blocks. Data is dynamically discovered at run-time via a registry that is distributed among multiple weather provider organizations. Data access services support on-demand filtering using spatial and temporal filtering attributes. The overall architecture is designed with flexibility in mind, as it is expected that the weather cube system topology will evolve significantly over time.

2.4.2 Gaps

NNEW architecture builds upon SWIM and FTI capabilities and shares the lack of traffic classification with those systems. In order for traffic classification to function properly, quality-of-service (QoS) hooks must be provided at the NNEW layer as well as at the SWIM and FTI layers. This work is currently ongoing in the NNEW program.

2.4.3 Impact on NWP Architecture

NNEW is designed from the ground up to support a variety of topologies and, therefore, places few constraints on the NWP architecture. Processor components that comply with the NNEW standards and services should be able to be distributed and/or centralized, and the overall partitioning of processing components should be easily modifiable over time. One impact of NNEW on the NWP architecture is that it may pay to decide up front the granularity of the functional blocks that may be distributed for segment 1 and/or future segments. The granularity that is chosen largely determines the placement and number of NNEW-compliant interfaces required.

3. PROCESSING AND COMPUTING TRENDS

When weighing candidate NWP architectures and possible implementation alternatives, it is useful to examine the types of processing required and current trends in processors and large scale processing systems. Where are processing performance gains likely to be centered in the coming years? How will the programming model change to accommodate changes in hardware architecture? Lastly, what techniques and technologies are available to manage and maintain all this computing horsepower? This section discusses some of these trends along with possible impacts and recommendations for the NWP.

3.1 WEATHER PROCESSING CLASSES

Weather data processing can be broken down into a number of different classifications, each of which can place different demands (i.e., requirements) on underlying compute hardware. The major categories typically encountered include

1. ***High-speed signal processing of weather sensor data.*** A/D samples from sensors are processed, producing fundamental output parameters such as reflectivity and velocity. Fast Fourier transforms (FFTs), high-speed I/O, real-time performance. Memory need – medium (sensor range). Up until the relatively recent past, this has been solely the realm of custom hardware and special purpose processors. In the past 10 years, the trend has increasingly been to use a mix of field-programmable gate arrays (FPGAs) and more general-purpose processors. The FPGAs tend to buffer the general-purpose processor from submillisecond real-time requirements, as well as provide a cost-effective solution for front-end processing in large multichannel radar systems (such as the Multifunction Phased-Array Radar (MPAR). Note that this category of processing is not being included in the NWP Segment 1 time frame and is not discussed further in this study.

2. ***Weather model computation.*** Raw computation and I/O are the driving requirements in this category. General-purpose processors are up to the compute task. To achieve the desired computation/IO balance, specialized I/O interconnects are often used. These interconnects handle much of the I/O work, offloading the general-purpose processor and allowing it to focus on core computation.

3. ***Image processing.*** Matched filters passed over potentially large CONUS-sized image fields. Pattern recognition and cross-correlation tracking fall into this category. High-memory throughput and a relatively large cache are keys to good performance in this category.

4. ***Product generation.*** This category is probably the most general-purpose and includes the post-processing of imagery to detect features such as microbursts and gust fronts. This tends to be a general mix of operations that requires an even balance of computation and memory performance.

5. ***Data I/O.*** I/O processing tends to reside at the edge of the core data processing components and is comprised of the processing duties associated with data formatting, data compression, and potentially encryption. This category, though relatively light in terms of required central processing unit (CPU) cycles, is important to the overall architecture since it exists at the boundary between processors and the rest of the system.

With the exception of the first category, the NWP processing architecture must be flexible enough to support all these processing classes. Key questions include when to use different physical hardware to support the needs of the different processing classes and when to target a more common hardware platform for maintenance reasons.

3.2 MULTICORE PROCESSORS

Over the past five years, there has been a significant shift in focus with respect to CPU design. Increases in clock speed, increasingly difficult to achieve due to power consumption, heat dissipation, and manufacturing issues, have given way to multiple cores as the primary path to increased processing power within the same overall footprint. In the span of only a few years, multiple cores have now become the rule rather than the exception, even on relatively low-end processing platforms such as laptops.

In the past, a high-performance processor might have been designated as a *symmetric multiprocessor* (SMP), with multiple cores (typically no more than 64 cores) accessing a common memory space, or a *compute cluster*, with a potentially large number of single cores (~100–100000) connected together in a networked topology. In order to exploit the power of an SMP machine, applications are typically multithreaded, often with the help of a support library such as OpenMP. In order to exploit the power inherent in a large cluster, applications are partitioned in such a way that the I/O between compute nodes is minimized to the extent possible, and common APIs like the Message Passing Interface (MPI) are used to minimize the interprocessor communications programming effort required for the individual software developer.

With the advent of multicore technologies, clusters now present a hybrid mix of SMP and conventional single-node hardware environments. Along with the increased horsepower comes an associated increase in programming complexity to exploit the available processing cycles. Unfortunately, software programming models to address this hybrid hardware model are not currently mature when compared to the hardware itself [1]. A version of OpenMP (Cluster OpenMP) targeted at the hybrid environment does exist, but it is not clear if it is seeing wide adoption. Likewise, there is ongoing discussion in the MPI community as to how to adapt MPI to the changing hardware landscape, but no clear direction as yet.

The problem of the hybrid programming model was confronted during the recent reengineering effort for the CIWS prototype. The current CIWS implementation is philosophically aligned with the Cluster OpenMP approach, providing a common API to threads on a local host (SMP model) or threads on a remote host (cluster model). The implementation, however, pre-dated Cluster OpenMP. Given that software support for clusters based on multicore technology is an active area of research, we recommend that this topic be investigated further prior to actual NWP implementation.

3.3 HIGH-SPEED INTERCONNECT FABRICS

The world of very high-performance computing uses a mix of interconnect fabrics, though the number of different interconnects is dropping over time as the interconnect, too, becomes a commodity item. A table comparing the types of interconnects used in the top 500 supercomputers for 2008 and 2009 is shown below. As shown in Table 2, the two dominant interconnects by far are Gigabit Ethernet (GigE), and Infiniband. Infiniband tends to be used where very high speeds coupled with low message latency is critical to the application, while Gigabit Ethernet is a lower cost, more ubiquitous technology that addresses applications that are more compute bound than low-latency I/O bound.

TABLE 2

High-Performance Computing Interconnect Types [2]

Interconnect	June 2008	June 2009
Myrinet	2.4	2.0
GigE	56.6	56.4
Infiniband	24.2	30.2
Proprietary	8.2	8.4
Other	8.6	3.0

A primary difference between Infiniband and Gigabit Ethernet is the ability of the Infiniband hardware to provide Direct Memory Access (DMA) from one physical machine to another, offloading the CPU cores from I/O interrupt handling chores. This has a significant effect on the scalability of a system as more physical servers are added. As the number of servers and associated message traffic increases, an Infiniband-based cluster is able to scale more linearly, making fuller use of the raw compute power of the CPUs than the GigE-based solution. This is shown in Figure 4. Note that this performance profile applies to the case of a numerical weather prediction model with demanding low-latency communications requirements. The scalability of a GigE-based cluster may be perfectly acceptable for a different type of processing task.

It should be noted that Ethernet-based technologies are not standing still. 10 Gigabit Ethernet is an emerging technology that is expected to gain market share in the high-performance arena in the coming five years. Whether it replaces one or both of the GigE and Infiniband technologies remains to be seen.

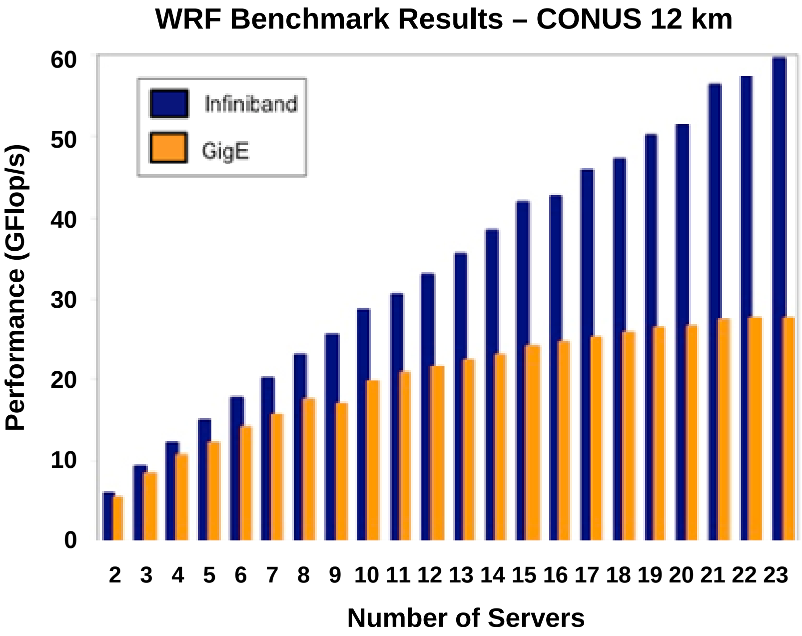


Figure 4. Scalability of Infiniband versus Gigabit Ethernet for weather model computations [3].

3.4 GRAPHICS PROCESSING UNITS AS APPLIED TO HIGH-PERFORMANCE COMPUTING

As weather models are run at increasingly high resolutions, they require correspondingly more and more compute resources. If general-purpose processors are used, the power consumption alone for very large clusters can run into the millions of dollars per year. An alternative emerging approach is to use more specialized processors for model computations, such as those provided with high-performance graphics cards. For floating-point intensive applications, coupling a graphics processing unit (GPU) with a more general-purpose process can result in a speedup factor of 25, or even higher. This is a very active area of research, and the graphics chip companies are working with the scientific community to add important features such as double-precision arithmetic to enable the chips to be used in these other applications. Similar to the case for multicore architectures, writing efficient code for these special-purpose processors requires detailed knowledge of the underlying architecture, as it stands today. This complexity should be reduced over time as libraries and programming best practices emerge from the scientific community.

From an NWP perspective, GPUs are interesting when thinking about advanced weather products that may depend on running high-resolution models. For the less demanding processing tasks that require 100s of CPUs rather than 1000s, the benefits of the more general programming models associated with traditional CPUs currently outweigh the costs.

3.5 VIRTUALIZATION TECHNOLOGIES

Virtualization refers to the ability to set up virtual machines that are decoupled from the underlying physical hardware. The basic ideas behind virtualization are not new, but in the past five years, virtual machines are increasingly becoming a part of large processing and/or web serving solutions. Benefits of virtual machines include

- ***Efficient use of servers.*** Servers, especially modern multicore servers, are often underutilized. The ability to run multiple instances of a virtual machine on a single physical server allows system administrators an additional degree of freedom to tune a processing cluster, without requiring software changes at the application or operating system level.
- ***Fault-tolerance.*** In large processing systems, some hardware failures are inevitable. Utilizing a virtual approach, stand-by hardware can quickly be configured to match the failed virtual resource and brought online.
- ***Ease of system management.*** Virtualization technologies allow for simplified management of large numbers of nodes. Software updates, for example, can be made to a single virtual machine master image, which is then propagated to any number of physical nodes.
- ***Support for multiple operating systems.*** The ability to run multiple operating systems on the same physical hardware is often useful for software developers and end users. This capability is less useful in the context of a large processing cluster – there is typically no need to run multiple operating systems in an environment focused on real-time data product generation. One exception to this rule exists in the cyber security area, where applications that are particularly sensitive may be hosted on more than one operating system to provide redundancy in the case of a security breach targeted at one operating system. This is likely not a requirement in the case of the NWP.

There are a number of approaches to virtualization, ranging from software emulation of one processor type on a different processor type to more lightweight options based on a “hypervisor” that resides between the operating system instances and the physical hardware. In the hypervisor model, nonprivileged guest operating system instructions are executed directly on the processor with no intervening translation step, resulting in an efficient use of processor resources. Only when privileged instructions, such as those associated with device I/O, are executed are the instructions intercepted by the hypervisor and mapped from their virtual to physical equivalents.

In order to minimize virtualization overhead even further, most newer 64-bit chip sets have built-in hardware support to accelerate virtual-to-physical resource mappings. As a result, the overhead associated with virtual machines can be very low, even negligible, for compute-bound applications. For applications with more significant I/O requirements, this is not necessarily the case, since I/O resources are potentially being shared by multiple virtual machines. This bottleneck can be reduced by the addition of multiple physical I/O devices to better match the number of virtual machines, though this obviously adds cost to the system.

Virtualization is not commonly used in the core of high-performance computing applications today. Rather than use virtualization as a technique to increase the usage of computation resources, high-performance applications are typically hand-optimized to extract the maximum performance out of the hardware. The high operational costs associated with large processing clusters (>1000 nodes) tend to make any optimization efforts along these lines worthwhile.

Perhaps the most promising role for virtualization technologies in the context of the NWP is in support of fault tolerance. Some virtualization solutions provide support for live migration of a software application from a faulted virtual processor to a replacement in a relatively transparent fashion. For mid-sized clusters that produce live weather products (as opposed to arguably less critical longer range forecasts), reliability and automated recovery is a critical issue that has not yet been addressed. This area is a good candidate for further research.

3.6 DATA CENTERS AS COMMODITY ITEMS

Commoditization of compute hardware does not end at the level of individual servers, or even racks of blades. Entire data centers are now becoming available as pre-package modules than contain much of their own power distribution and cooling infrastructure. They need only to be shipped a customer’s site and hooked up to the appropriate power and chilled water connections and they are ready to go. An example of such a system, Sun’s Modular Datacenter, is shown in Figure 5 below.

If a customer site already has an in-house computing center with expandable, maintainable power and cooling infrastructure, then this approach may not be cost effective. It can be very useful, however, for cases where a data center needs to be installed in a very short time frame, such as may be the case for web-focused startup companies.

We do not necessarily see a role for a “cluster-in-a-box” solution in the NWP Segment 1 time frame. There may be a role, however, for a commercially supported cluster solution, whereby an IT-focused company provides and maintains the processing platform and weather-focused organizations provide the software that is hosted on that platform.

Figure 5. Sun Microsystems modular data center.

3.7 CLOUD COMPUTING

Large Web-focused companies such as Amazon, Google, and Yahoo have developed scalable, robust compute infrastructures over time. Within the past several years, a new business model has emerged whereby these companies and others sell the infrastructure itself as a product, essentially leasing compute power and storage space on demand to clients who would otherwise find it difficult to stand up their own scalable processing environment. In this leased, on-demand model, the computing resources reside, in effect, "in the cloud."

The concept of leasing compute cycles on an on-demand basis rather than buying the processors outright introduces interesting additional alternatives for the NWP. Whether or not the software executing on the processor hardware is developed and/or managed by the FAA, the National Weather Service (NWS), or a commercial weather vendor, the possibility exists to use commercial IT resources to manage and maintain the processing hardware. In this model, scaling the hardware either up or down becomes relatively straightforward. Research partners, rather than standing up their own infrastructure, could similarly lease the compute cycles needed to perform the necessary computations and terminate the lease when the R&D product either moved to the operational environment or reached the end of its R&D evaluation period.

A key question for FAA usage is whether or not the leased approach could provide the quality of service required by the aviation community. Given the recent emergence of this approach, additional study is needed to determine if it is a viable option for the NWP Segment 1 time frame.

4. OVERVIEW OF EXISTING WEATHER SYSTEMS TO THE NWP ARCHITECTURE

Three key systems, WARP, ITWS, and CIWS, have been identified as the primary FAA-hosted weather processing systems that are affected by the transition to the NWP. This section provides background on the three systems for subsequent discussion of NWP alternatives. The processing subsystem of a fourth system, Next Generation Weather Radar (NEXRAD), is also described as the cross-agency software development and deployment model is relevant to the NWP. The inclusion of information for additional systems with a weather processing component (i.e., Aviation Digital Data Service (ADDS)) is reserved for more detailed follow-on studies.

4.1 WARP

4.1.1 Overview

The WARP system provides weather data to a variety of users including air traffic controllers (via Display System Replacement (Enroute) (DSR) and En Route Automation Modernization (ERAM)), air traffic supervisors, and command center personnel. WARP also disseminates data to a variety of other systems, including Advanced Transport Operating System (ATOP), User Request Evaluation Tool (URET), Dynamic Ocean Tracking System (DOTS), and ITWS. The primary sensor for WARP is the NEXRAD radar network, though it also ingests and disseminates weather station data, lightning data, and satellite data. Instances of the WARP system exist at the 21 ARTCC facilities around the country, as well as at the FAA command center.

A block diagram of the WARP system is shown in Figure 6. The system is decomposed into a number of subsystems, including

- Radar Acquisition and Mosaic Processor (RAMP). This subsystem is responsible for acquiring data from the NEXRAD radars, generating regional and CONUS radar mosaics, and outputting the mosaic products to the air traffic controller displays (DSR/ERAM) and the WINS subsystem.
- Weather Information Network Server (WINS). This subsystem is a general-purpose data aggregation and dissemination system.
- WARP Briefing Terminal (WARP BT). This subsystem provides a means for presenting weather information to supervisors and meteorologists residing at the ARTCC facilities.

Of these subsystems, the most relevant to the NWP is RAMP, since it is the only subsystem doing data processing. WINS is also relevant since it is naturally aligned with the NNEW and SWIM functionality upon which NWP builds.

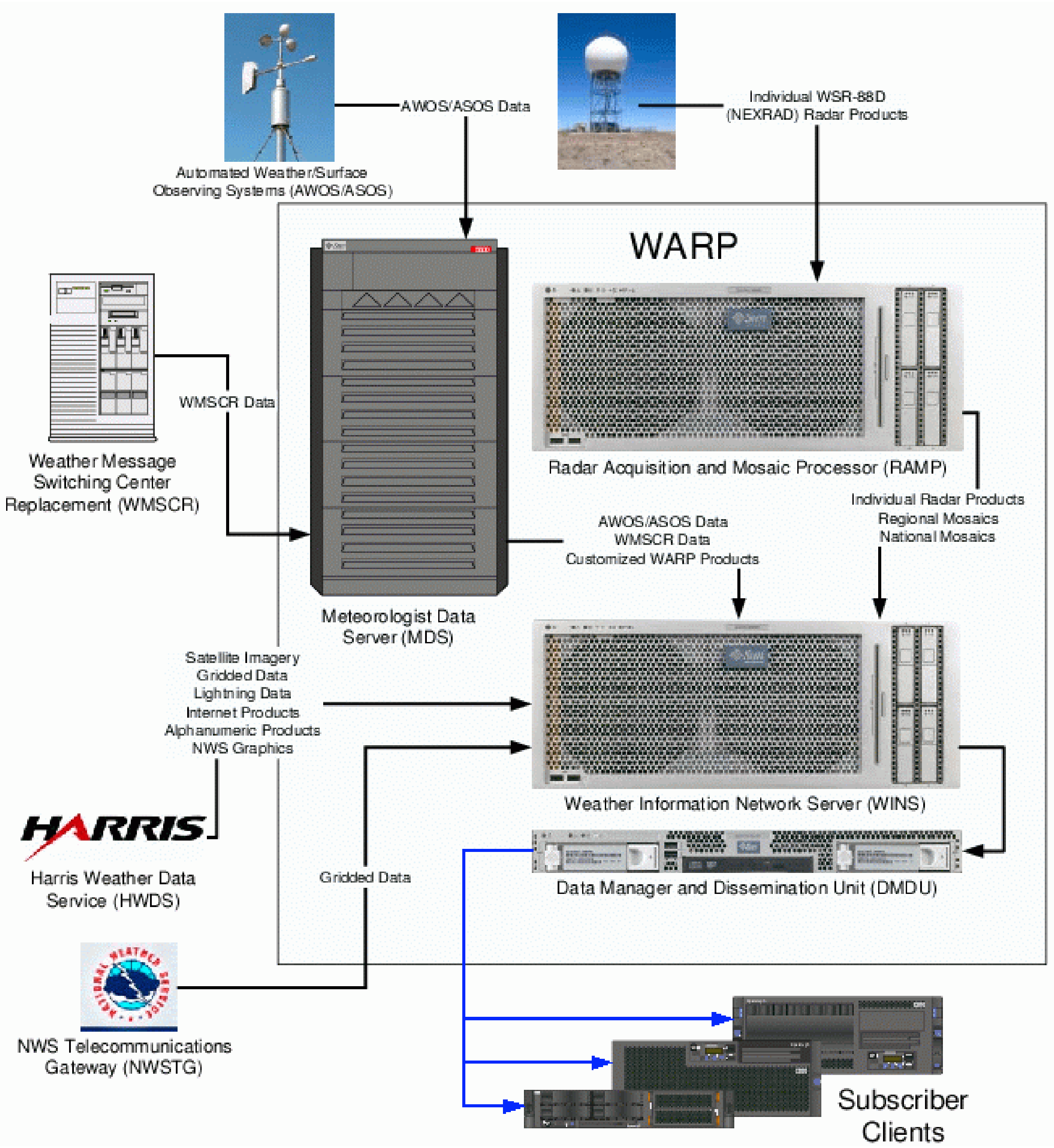


Figure 6. WARP design following technical refresh [4].

4.1.2 Alignment with NWP Architecture

The WARP architecture naturally aligns with the NWP architecture in a number of respects. Like NWP, it supports a mix of distributed and centralized components. Data acquisition and dissemination is largely separated from the processing subsystem, with the exception of the NEXRAD data acquisition and dissemination to the DSR/ERAM displays. Since WARP resides at ARTCCs, it naturally aligns with the FTI infrastructure, since each ARTCC is a node on the FTI backbone.

4.1.3 Challenges and Opportunities Associated with Transition to NWP

Opportunities associated with the transition of WARP processing functionality to the NextGen model exist in the product improvement realm as well as the IT realm. On the product improvement front, an opportunity exists to introduce motion compensation into to the radar mosaic generation process, improving the spatial accuracy of the product and eliminating spatial "jitter" associated with asynchronous single radar updates to the mosaic. In order to support this approach, one minute or 30 second updates to the motion compensated products will likely be required to better match the 25 second update rate of the existing WARP system, as opposed to the 2.5 minute updates being used today in the CIWS system. This not only matches up better with existing WARP system, but also with the nominal 30 second update rate for weather products produced by terminal area radar systems (ASR-9/11).

A challenge associated with the motion-compensated radar mosaic product is user acceptance, especially in the NWP Segment 1 time frame. An approach for NWP Segment 1 may be to provide the new product alongside the existing product at one or more key sites, in preparation for a full switchover in NWP Segment 2.

In an environment where NEXRAD data is available in the NAS with the required latency and reliability characteristic via NNEW, SWIM, and FTI, the RAMP functionality is more naturally implemented using a centralized approach than today's distributed approach. Unfortunately, the elimination of the mosaic function in each ARTCC does not eliminate the need for the RAMP hardware, since RAMP drives the DSR and ERAM displays, and the interface is not IP-based. It is not yet known if ERAM is planning to directly support an IP-based, NNEW-compatible interface to controller displays. If this is not the case, then ERAM tasking should be modified as needed to remove the need for the current custom interface in the long term.

From an IT perspective, the recent technology refresh of the WARP system presents integration opportunities as well. The use of commodity hardware, as well as software that is well aligned with NNEW and SWIM infrastructure lowers the barrier for migration of existing WARP functionality to the NextGen model [4]. In particular, the adoption of publish/subscribe technologies (JMS) for the WINS subsystem should greatly simplify the construction of NNEW-compliant service. One minor challenge to this approach is, as mentioned in Section 2, the lack of an on-the-wire standard for the JMS publish/subscribe technology. Barring a wholesale replacement of the WARP MQ/Series JMS implementation with the SWIM ActiveMQ product, some bridging between JMS implementations will likely be required.

4.2 ITWS

4.2.1 Overview

ITWS serves the nation’s major airports, generating aviation weather products that are displayed on dedicated displays for FAA users in Control Towers, TRACONs, and ARTCCs. These same products are provided to users in the command center via Intranet using a website that is part of the existing production system. Airline operations centers, and other approved users, may view ITWS information by accessing this same website over dedicated connections to Volpe or they may receive ITWS product digital data through a SWIM Segment 1 interface that is currently under development. Pilots receive ITWS information through an uplink of special (Terminal Weather Information for Pilots (TWIP) messages to the cockpit. The products are intended to improve both the safety and efficiency of airport operations during adverse weather.

ITWS ingests weather data from a number of FAA and NWS radars and sensors, including the TDWR, NEXRAD, ASR (Models 9 and 11), Low Level Windshear Alert System (LLWAS), Automated Weather Observing System (AWOS), and the Automated Surface Observing System (via ADAS). Other inputs include the National Lightning Detection Network, NWS Rapid Update Cycle data, and the Meteorological Data Collection and Reporting System. These data are integrated to produce products that are intended to be used directly by the FAA users, without requiring meteorological interpretation. These products range from the warnings of potentially hazardous weather (microbursts, windshear, gust fronts, hail, lightning, and tornadoes) to portrayals of current (precipitation mosaics, echo tops, storm cell motion, and terminal winds) and predicted weather (up to one hour, showing both convective and winter weather).

A block diagram of the ITWS system is shown in Figure 7.

4.2.2 Alignment with NWP Architecture

In terms of the general NWP architectural vision, which contains both centralized and distributed elements, ITWS lacks centralized processing capability. There are a couple of minor exceptions to this claim. In the areas of data acquisition, ITWS has a National Filter Unit (NFU), which has limited filtering and slicing and dicing capabilities to forward national data to the individual ITWS processors. In the area of data dissemination, the ITWS products that appear on the dedicated displays are forwarded to Volpe for incorporation into the centralized ITWS website and for use as input to ITWS SWIM prototype for product dissemination to external users.

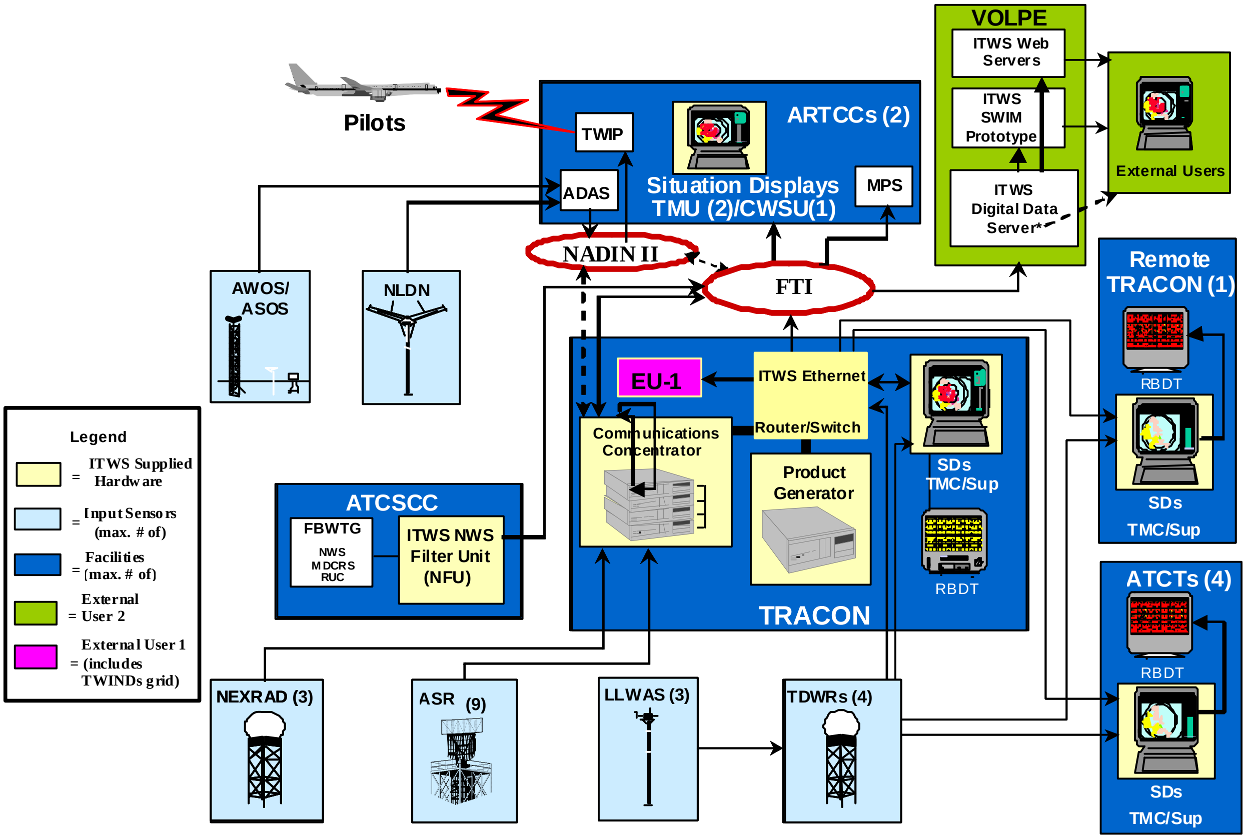


Figure 7. ITWS design including the VOLPE SWIM Segment 1 prototype. Not depicted in this drawing is the access by the ATCSCC users to the ITWS website from Volpe. Though shown as a possibility in the drawing, the direct access to the ITWS digital data has not been used directly by external customers, only as a feed to the SWIM prototype.

By its nature, ITWS does offer the opportunity to create a local data concentrator. In addition to the variety of data sources it acquires to create its products, ITWS has local access to the TDWR data, which could be leveraged in the NWP Segment 1 time frame. This may be particularly important since it is unlikely the latency issues that affect ITWS's need for timely high-bandwidth TDWR data will be resolved before NWP is first deployed. As a footnote on the data ingest discussion, most of the data moves to and from ITWS over network connections. There are, however, three dedicated connections for the ITWS: NEXRAD, ASR, and TDWR.

The level of decomposition between data acquisition/dissemination and processing is similar to WARP's. For the most part, all data move in and out of ITWS through the communications concentrator; however, in the case of the TDWR, which would have overwhelmed the I/O handling capabilities of the concentrator hardware, the data are acquired directly through the Ethernet interfaces on the product generator itself.

One concern, when looking at ITWS architecture from a functional perspective, is the tight coupling between an ITWS product generator and its dedicated displays. This is particularly important in that the TRACON and Tower displays double as input devices, transmitting airport runway configuration decisions and pilot provided windshear reports to the ITWS product generator that affect the ITWS products themselves.

4.2.3 Challenges and Opportunities Associated with Transition to NWP

The current ITWS processing architecture is based on a Symmetric Multiprocessor approach. It was implemented using a Sun Fire 3800, no longer available from Sun, and it presumes that 32 bit operation is all that is required. As part of the original deployment, the transition from 32 to 64 bit processing was deferred. At the time, it was recognized that some development (and certainly considerable testing) would be required, which needs to be factored in should this become a starting assumption for NWP. ITWS does have a tech refresh planned, but it would have to be significantly accelerated in order for ITWS to be considered as part of the near term transition plan for NWP. The system is not inherently scalable beyond a single box and the current system is limited to 8 processors, which, although adequate for the existing ITWS applications, is not a suitable starting point for meeting the NWP requirements.

From a software perspective, a proprietary Raytheon operating system overlay (NOS) provides message passing and process management system support for the ITWS, which introduces obstacles to transition, both from the perspective of transparency and from the view of configurability. NOS requires static memory and buffer allocations, as well as custom adjustments of priorities in order for competing processes to meet their latency requirements. Making adjustments to existing configurations is both labor intensive and requires specialized expertise. In addition, a new approach to system control would be needed to decouple the display from the product generation. That having been said, if decoupled, some replacement for access to the users' information, which portions of the ITWS algorithms require, would have to be developed.

In terms of the ITWS current dissemination architecture for the ITWS products, it may be practical if one adopts the view that internal users have access to ITWS via the ITWS displays and that a central gateway will be all that is needed to support external clients. Since the data were already aggregated at a central location to support the ITWS website, it was difficult to argue against extending this architecture in the near term to centralize access to the data for external users of SWIM. However, this implementation has its detractors; going forward, there has been discussion about moving the publication closer to the source, in recognition that ITWS is fundamentally a distributed product generator. Regardless of the dissemination architecture that is ultimately used for these products, there is one product (the ITWS Terminal Winds Grid), which although likely to become a very desirable input to traffic flow optimization algorithms of the future, has not been provided to clients except through the simple text profile summary that appears on the dedicated displays. This product is available only the EU-1 interface shown in the diagram, and as such is not sent to the Volpe as the ITWS SWIM service point. In any case, this omission should be addressed.

4.3 CIWS

4.3.1 Overview

The CIWS system fuses data from a variety of sensors and weather models and produces a high-resolution, high update rate forecast for the ATC community. Figure 8 provides a block diagram of the major subsystems, including data ingest, cluster-based processing, and product output. Inputs include satellite data, lightning data, imagery from ground-based radars, winds data, and weather model data. With the exception of the lightning and satellite data feeds, which are based on satellite downlink, all products are received via the Internet (some via Internet 2). Data is disseminated to ATC users with dedicated displays over a private frame relay network. Airline and other users access CIWS imagery via a Web portal over the commodity Internet. A recent addition with respect to CIWS data dissemination is the SWIM CIWS Data Distribution Service (CDDS), an NNEW-compatible service whose major components are shown on the right-hand side of the diagram.

4.3.2 Alignment with NWP Architecture

The CIWS prototype is based on a modular design that cleanly separates core processing functionality from data I/O. It is based on a purely centralized processing model, using a single cluster based at Lincoln Laboratory. Based on a commodity Linux cluster, the processor is highly scalable, having been scaled up numerous times throughout its lifetime. The scalability of both the CIWS and software is the key property of the system of interest to NWP, since neither WARP nor ITWS is scalable to the same extent. Though CIWS does not inherently support the notion of distributed processing, the follow-on Consolidated Storm Prediction for Aviation (CoSPA) effort does, in fact, support multiple processing locations.

Data I/O is generally aligned with the NNEW hub and spoke model. On the ingest side, data for a number of sensors is received using Unidata’s Local Data Manager (LDM), which is itself based on a hub and spoke approach. On the data dissemination side, repeater processes installed at ARTCC facilities to “fan-out” data to multiple users at each facility. The more recent CDDS subsystem is more than aligned with the underlying SWIM/NNEW architecture, since it actually *conforms* to the specified data standards and service interfaces defined by those programs.

4.3.3 Challenges and Opportunities Associated with Transition to NWP

The CIWS system currently experiences some regional reliability issues, primarily due to the regional outages in availability of the NEXRAD level II data. These issues will be addressed sometime in 2010 with the installation of additional backup of level III products. In addition, the CIWS system itself does not have a high level of hardware redundancy and automatic fault tolerance. Though failures of CIWS hardware are relatively uncommon, staff resources in the form of on-call personnel are required to reconfigure resources to achieve the relatively high uptime statistics for the system. This issue will need to be addressed for components of the NWP that are inherited from CIWS. A combination of redundant hardware and virtualization is a possible approach and is a candidate for further research.

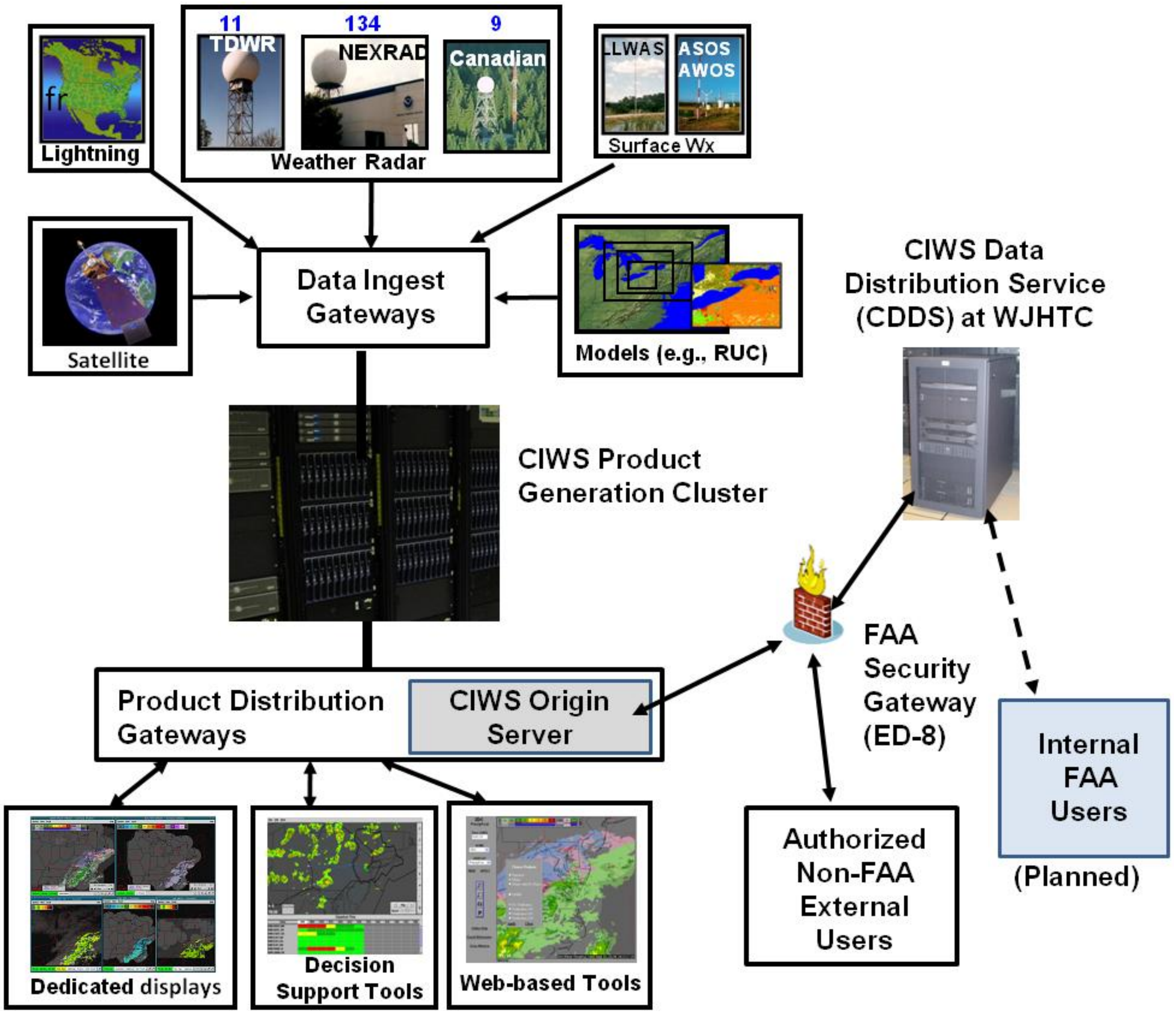


Figure 8. CIWS architecture, showing CIWS prototype. The prototype includes the primary CIWS engine and CIWS origin server, which operate at MIT Lincoln Laboratory and the CDDS node, which is part of SWIM Segment 1 and operates from WJHTC behind the FAA's firewall.

As mentioned in Section 3, modern clusters based on multicore technologies present programming challenges with respect to achieving high utilization of the hardware, while at the same time keeping the software simple. CIWS utilizes a custom programming model based on the ideas in OpenMP (an SMP-based approach), but CIWS development pre-dated work on a version of OpenMP targeted at the cluster environment (Cluster OpenMP). Assessment of the state of the art in high-performance computing programming models, including Cluster OpenMP and other models, should take place prior to transition of CIWS software to an NWP environment.

CIWS presents significant opportunities for consolidation and improvement of weather products. A CIWS-based motion-compensated mosaic is a good candidate to eventually replace the WARP regional and national mosaics. The CIWS convective weather forecast is a candidate for replacing older functionality embedded in the ITWS system. Consolidation of the different mosaics and forecasts to be based on CIWS (eventually CoSPA) is in keeping with the idea of a single authoritative source for a given weather product and helps to achieve the common operational weather picture that is part of the overall NextGen vision.

4.4 NEXRAD OPEN RADAR PRODUCT GENERATOR

The NEXRAD processor architecture is not specifically mentioned in the NWP alternatives as one of the candidate contributing architectures, but it is interesting in one particular respect – the lifecycle and development model used for the product generator. The NEXRAD approach is to provide an open processing platform, allowing multiple agencies to develop and deploy data products tailored for their purposes without requiring a heavyweight technology transfer process. The Open Radar Product Generator (OpenRPG), currently based on a commodity dual-processor personal computer (PC) running Linux, provides an API that provides standard methods for accessing raw radar input data and outputting derived products. The software environment is made available to requesting organizations and is capable of running on a variety of PCs.

Using the OpenRPG model, the path to a deployed product typically includes offline development of the product using an in-house instance of the OpenRPG and archived data followed by handoff to the NWS for testing and final deployment of the product generation algorithm. Although the NWS is responsible for verifying that the algorithm operates as expected, and the product is reliably produced, the detailed knowledge of the algorithm’s inner workings is often maintained within the agency that originally developed the product. The NWS role is to provide the processing platform, not necessarily take responsibility for the algorithm itself.

The OpenRPG approach has proved very successful, allowing insertion of a number of products of interest to aviation over the past decade. The Machine Intelligent Gust Front Algorithm (MIGFA), originally developed for the TDWR radar system, was successfully implemented via the OpenRPG technology insertion path. There are signs, however, that the distributed OpenRPG processing model may be a limiting factor, as algorithms grow in complexity and increasingly turn to data fusion to improve data quality. Data fusion requirements tend to drive designs to a more centralized approach, assuming that the appropriate communications framework is in place, but the decision as to where to draw the line is not always clear.

The centralized analog to the OpenRPG model for the NEXRAD community is to provide a more centralized, scalable computing resource that allows different organizations to request the necessary compute bandwidth and install software algorithms for a burn-in test, followed by a transition to operational use. This is, for all intents and purposes, a common goal with the NWP and is very closely aligned to the cloud computing model (compute resources on demand) described in Section 3. Figure 9 illustrates the relationship between the OpenRPG and an NWP processing infrastructure implemented along similar lines.

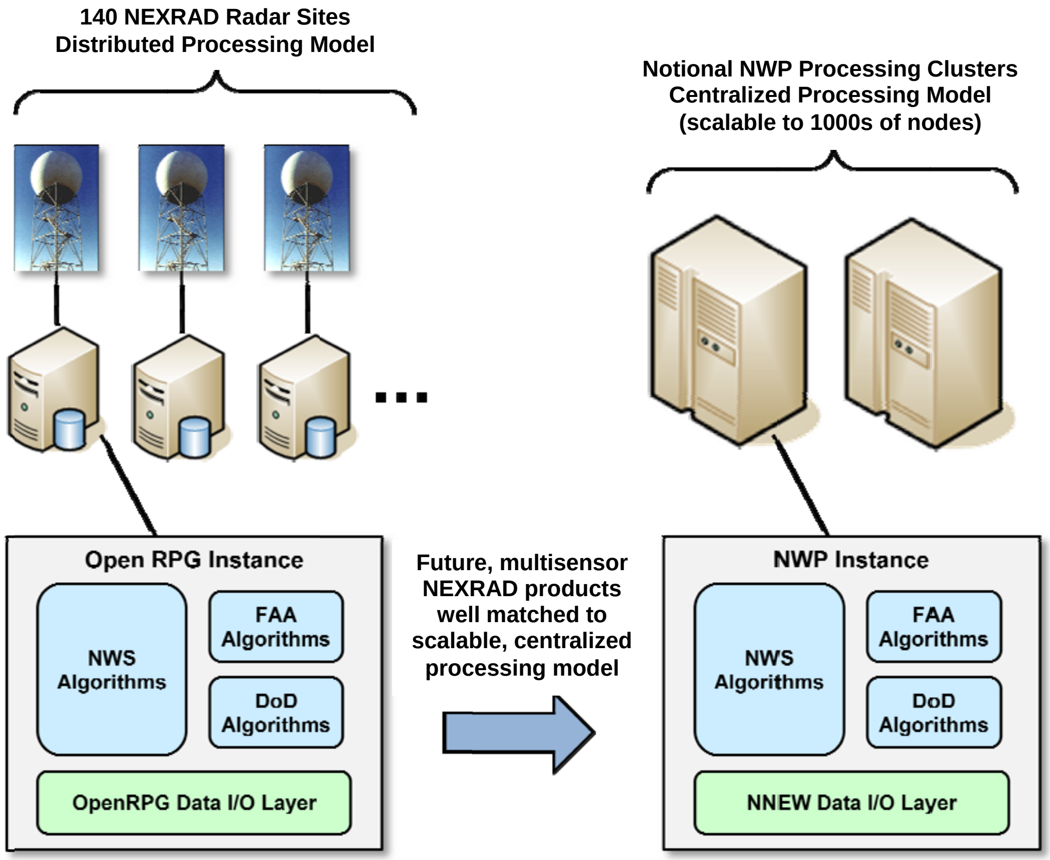


Figure 9. Open RPG weather product development and deployment model has been highly successful at single-radar scales, but has limitations with respect to scalability and algorithms that depend on multisensor fusion. A similar "provide the processing infrastructure" approach at cluster scales would maintain the current multiagency development agility and provide the necessary scalability.

5. NEXTGEN WEATHER PROCESSOR HIGH-LEVEL REQUIREMENTS AND ARCHITECTURAL GUIDANCE

In order to better assess NWP implementation alternatives, it is useful to have a high-level picture of the NWP requirements and architectural principles that help address the requirements. This section describes the requirements in an abstract sense, compatible with the long-term NextGen vision. Architectural guidance consistent with those requirements is also presented.

5.1 REQUIREMENTS

Most distributed systems support a primary, system-specific objective and share a similar set of secondary objectives such as *security, agility, reliability, maintainability,* and *affordability,* the so-called "ilities." By way of example, consider the following NWP Segment 1 primary and secondary objectives.

> ***Primary Objective:*** Support legacy WARP/ITWS/CIWS capabilities, and advanced weather capability.
>
> ***Secondary Objectives:***

1. ***Latency.*** Products must be made available to consumers within specified time constraints.
2. ***Reliability.*** Data that is not available at the desired level of service is not useful.
3. ***Security.*** The NWP architecture must support secure access to data in accordance with the policies of the FAA, NWS, and commercial organizations.
4. ***Agility.*** The NWP architecture must support change over time with regard to distributed versus centralized components, shifts in organizational responsibility for a given product, and changes in processing technologies over time.
5. ***Scalability.*** The NWP architecture must be scalable, both in the increasing and decreasing directions (to accommodate shifts in partitioning of functionality). This objective is closely related to the more generic agility objective.
6. ***Maintainability.*** The NWP architecture should encourage a design that is highly maintainable.
7. ***Affordability.*** The NWP architecture should result in a cost-effective design, taking advantage of off-the-shelf hardware and software processing capabilities wherever possible.

Many of the secondary objectives listed are in natural tension with one another. For example, a system with complex security policies will likely be less agile than a system with a more relaxed policy. Likewise, a system where affordability is judged more important than reliability might choose to avoid the expense of redundant compute hardware in order to meet the cost objective. In general, a simple unordered set of objectives provides little in the way of design guidance. When prioritized, however, the list of objectives tends to guide key architectural decisions [5].

This study assumes that the objectives are prioritized in the order shown above. A danger of prioritizing such a list is that to a security-focused reader it may imply that security will be sacrificed for reliability and latency. Similarly, given that affordability is ranked last, to a finance person it may seem equivalent to stating that "money is no object." This is not actually the case. In the case of security, for example, the prioritization simply indicates that the security solution must not negatively impact the desired latency and reliability objectives – those objectives take precedence. This obviously has an impact on the design of the security solution – it must necessarily be fast and efficient. Likewise, security is ranked higher than affordability, which implies that the resources needed to secure the system will need to be made available to satisfactorily implement the design.

5.2 NWP ARCHITECTURAL GUIDANCE

As discussed earlier, the NWP will leverage the NNEW, SWIM, and FTI infrastructure and inherits a significant amount of architecture from those programs. The availability of a location-independent 4-D data cube that acts both as a data source and data sink provides a good deal of system agility. The fact that the 4-D cube builds upon SWIM and FTI largely satisfies the security objective. The communications portion of the latency objective is similarly the collective responsibility of the NNEW, SWIM, and FTI infrastructure programs.

Processing on the scales most relevant to the NWP is to a large extent a commodity in today's world. The use of commodity hardware and software addresses a number of the maintainability and affordability objectives. The use of commodity processing elements also addresses the agility objective, in that a commodity cluster is easily scaled, and outdated processing components can easily be upgraded without requiring major architectural changes.

A key question is, given the amount of inherited architecture and the commodity status of processing clusters, what architectural decisions remain that are specific to the NWP? A number of candidate architectural decision points are described in the following sections.

5.2.1 NWP Hardware Profiles

Though processors are a commodity item, there is no one-size-fits-all processor that is optimally suited for all weather processing tasks. To be cost effective, some different configurations are generally necessary. The benefit of differentiating the hardware based on function must be balanced with the additional cost incurred to support multiple hardware types. In our CIWS prototype work, which does not include running of large data models, we have found that supporting two different processing configurations, one for CONUS-scale image processing and one for data I/O, is a practical approach. Table 3 below adds a third category to address the high-performance floating-point computation requirements of high-resolution weather data models. Supporting more than three weather processor profiles within an organization begins to incur maintenance costs that exceed the benefit of the tailored hardware.

TABLE 3

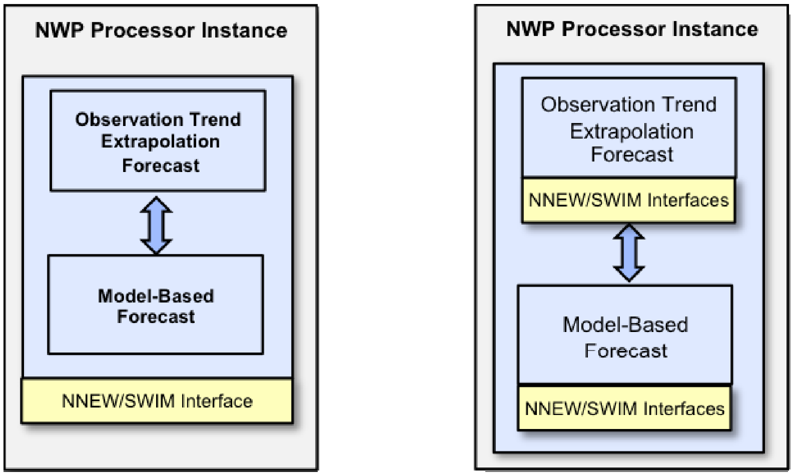
NWP Processing Hardware Profiles

Processing Category	Key Processor Characteristics
Weather Model	• High floating-point performance • Scalability to 1000s of nodes • Medium memory requirement per node • Very high-speed processor interconnect (10 Gbps)
Image Processing/ Product Generation	• Balanced floating-point and integer performance • Scalability to 100s of nodes • Large memory requirement per node • High-speed processor interconnect (1 Gbps)
Data I/O	• Integer performance (data movement) • Scalability to 10s of nodes • High memory size per node • Medium/high-speed processor interconnect (100 Mbps–1 Gbps)

The use of higher-end processor interconnects (e.g., Infiniband) promises higher performance, but unless absolutely required (typically the case for weather model processing) should be avoided for reasons of maintainability and cost.

5.2.2 Granularity of Weather Processing Functional Blocks

In order to benefit from the SOA-based infrastructure, functional blocks should be fine-grained enough to allow them to be composed in different configurations over time. Each functional block that is potentially useful as a stand-alone entity should conform to the NNEW/SWIM interface standards (Figure 10b) to allow it to be easily migrated to different nodes in the overall NWP processing infrastructure.



(a) Coarser-grained weather algorithm interface approach

(b) Finer-grained weather algorithm interface approach

Figure 10. Coarse versus finer-grained partitioning of weather algorithm functional blocks. In the case of b), the individual extrapolation and model-based forecast modules may be easily relocated to another location if needed, due to their support for the NNEW/SWIM interface standards.

5.2.3 Common Software Languages and Tools within a Functional Block

In order to promote maintainability, a common software infrastructure (e.g., image processing libraries) should be used within each functional block. Use of common software infrastructure across functional blocks should be a goal, but not a requirement, as there are valid reasons for supporting multiple languages and libraries. The use of FORTRAN in existing weather modeling applications, for example, should not preclude them from being run on the NWP. Where multiple major versions of a language exist, interoperability within a given language domain should be promoted by adoption of a common version wherever possible.

5.2.4 Layered Approach to Reliability

Similar to the concept of “Security in Depth,” we recommend a layered approach to reliability for the NWP. This includes the traditional use of redundant power supplies and available spares at each processor location. For the more critical and larger cluster-based processing nodes, this is traditionally augmented by one or more backup systems at separate geographical locations to prevent a single regional failure from affecting the entire CONUS for an extended period. This is a practical and proven approach that applies in the case of NWP.

Processing clusters have large numbers of nodes, and the chance of failure of a single node generally grows as the size of the cluster grows. In addition to a geographically diverse system, it is desirable to have some capability to automatically detect and recover from failures, or anticipated failures, at each redundant processing location. Though detection of failures has been implemented in the CIWS system, automated recovery and failover to a hot spare has not been demonstrated. The fault tolerant features supported by a number of processor virtualization solutions hold promise in this area, but additional work will be required to move the approach into a real-time processing cluster environment.

5.2.5 Location-Agnostic Processing

It is difficult to know in advance which locations and organizations will provide processing platforms for NWP Segment 1 and how that distribution may evolve in subsequent segments. The key for the architecture is to provide the necessary flexibility so that evolution of the system is easily accomplished over time. Adopting the cloud-computing paradigm, where processing is provided by an organization or vendor as a service, provides the flexibility required to meet the NextGen agility objective. As shown in the figures below, adoption of this approach allows all NWP stakeholder organization to host their own processing resources (Figure 11), or, alternatively, some organizations may choose to leverage processing provided by other stakeholders (Figure 12) to eliminate the need for a physical processing facility.

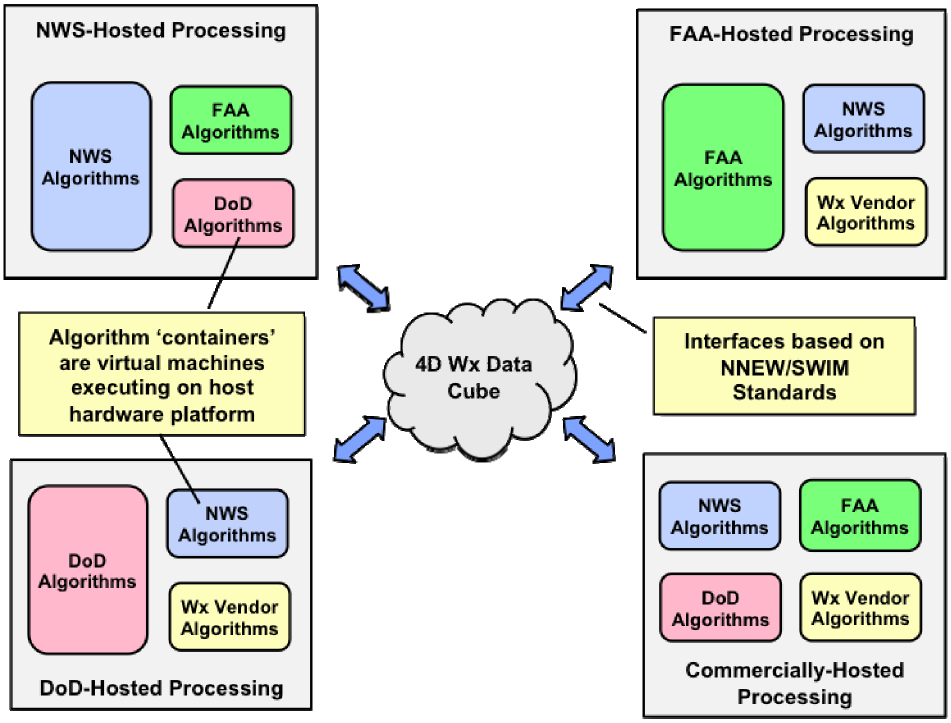


Figure 11. Sample instantiation of high-level NWP architecture. In this sample, NWS, FAA, DoD, and a commercial vendor all participate as members of a distributed processing group. Each member organization is provided computing resources at each location, depending on need.

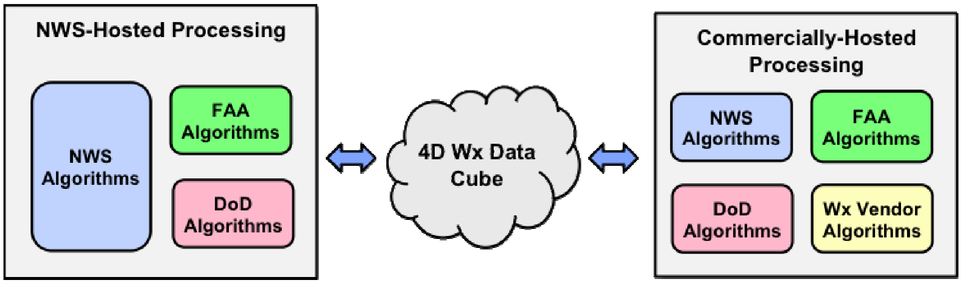


Figure 12. Variant instantiation of high-level NWP architecture. In this sample, only two organizations, NWS and a commercial vendor, provide processing infrastructure. The FAA and DoD run algorithms on the processors, but do not maintain their own capability, with the objective of reducing maintenance costs.

5.2.6 Layered Approach to Governance

The topic of governance is perhaps not commonly associated with architecture, but in order to support the flexible processing model described in the previous section, a well-matched approach to governance is needed. The FAA will not likely feel comfortable running algorithms on an NWS processor if there is not a measurable amount of agility built in to the governance model. In other words, if the FAA wants to install a new algorithm on an NWS processing host, other than helping test the algorithm to make sure it executes reliably and within the allocated resources, there should be little in the way of barriers to doing so. If this flexibility does not exist, then individual organizations are driven to maintain their own processing capability to provide the necessary flexibility, which can increase maintenance costs.

One alternative used by a variety of standards organizations is to use a composable, extensible governance approach, with a common base working group that manages algorithms of interest to multiple parties, and individual, smaller, working groups for the particular weather subdomains (NWS, FAA, DoD). The overall goal of the approach is to exploit cross-domain processing commonality where possible, while preserving an efficient pathway for individual organizations to innovate. This can be thought of as a layered approach to governance (Figure 13), with a lower cross-domain layer and an upper domain-specific layer. The lower layer governs the common algorithms and typically evolves more slowly, while the upper layer provides the necessary agility. As an algorithm matures and potentially begins to be used by other NWP stakeholders, the governance for the algorithm may move into the lower layer.

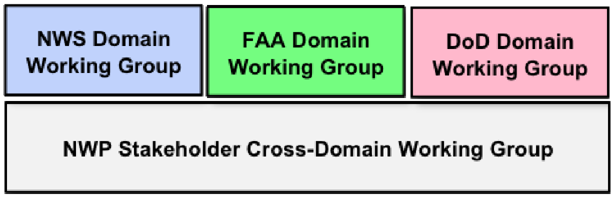


Figure 13. NWP layered governance model.

6. NEXTGEN WEATHER PROCESSOR ALTERNATIVES ANALYSIS

This section presents a high-level analysis of the currently identified NWP alternatives, building upon material presented in the previous sections.

6.1 ALTERNATIVES AND SUBALTERNATIVES

The NWP list of alternatives presented in Section 1, expanded to include the identified subalternatives, is shown in Table 4.

TABLE 4

NWP Alternatives and Subalternatives

FAA	**FAA produces advanced forecast products and legacy products. FAA publishes and subscribes to products**
	Subalternative 1. ITWS (modified) generates and publishes legacy products and advanced weather capability.
	Subalternative 2. WARP (modified) generates and publishes legacy products and advanced weather capability.
	Subalternative 3. CIWS (production version) generates and publishes legacy products and advanced weather capability.
	Subalternative 4. New NWP infrastructure provides advanced forecast and legacy capability. WARP and CIWS phased out. ITWS net-enabled.
NOAA	**NOAA provides advanced forecast products and optionally FAA legacy products**
	Subalternative 5. NOAA provides advanced forecast products. FAA provides legacy WARP, ITWS, CIWS products.
	Subalternative 6. NOAA provides advanced forecast products and legacy products.
Commercial Vendor	**Commercial vendor provides advanced forecast products and optionally FAA legacy products**
	Subalternative 7. Commercial vendor provides advanced forecast products. FAA provides legacy WARP, ITWS, CIWS products.
	Subalternative 8. Commercial vendor provides advanced forecast products and legacy products.

There are two primary questions embedded in these alternatives. The first is "Should the processing associated with the NWP be distributed between candidate organizations, and if so, how should it be partitioned?" The second question is "If the FAA chooses to do this in-house, how can the existing systems best be leveraged?" These questions are addressed separately in the following sections.

6.2 PARTITIONING OF NWP PROCESSING AMONG STAKEHOLDER ORGANIZATIONS

Five different processing partitioning approaches are implied by the alternatives. They include FAA-only, NOAA-only, and vendor-only solutions, as well as FAA/NOAA and FAA/vendor solutions. The high-level pros and cons associated with each approach are shown in Table 5. For the purpose of this study, it is assumed that the NOAA and vendor alternatives consist of "weather products as a service."

TABLE 5

Pros and Cons Associated with NWP Organizational Partitioning

	Pros	Cons
FAA-Only	• High-reliability network in place (FTI) • Leveraging of existing FAA systems • No need to bridge multiple networks for accessing low-latency safety critical weather products • Control over budget/schedule	• Possibly redundant weather processing data centers in U.S. Government • Large NWS-produced model data sets must be passed to the FTI network through the ED-8 gateway • Not a "pure" NextGen approach in terms of seamless data sharing between organizations
NOAA-Only	• Consolidated weather processing data center(s) for U.S. Government • Elimination of need to maintain FAA-specific weather processing systems	• Reliability and latency characteristics of NOAA network • Requires strategy for handoff of FAA R&D to NOAA, as well as ongoing governance • Lack of control over NWS budget/schedule
Vendor-Only	• Leverage private sector R&D and operational capabilities • Elimination of need to maintain FAA-specific weather processing systems • Control over budget/schedule	• Reliability and latency characteristics of vendor network • Single weather vendor lock-in • Possible restrictions on use of data introduced. This is counter to the open and accessible NextGen 4-D cube philosophy

	Pros	Cons
FAA/NOAA	• High-reliability network in place for legacy products • Leveraging of existing FAA systems • Partial control of budget and schedule	• Possibly redundant weather processing data centers (maintenance issue) • Reliability and latency characteristics of NOAA portion of network • Requires strategy for handoff of FAA R&D to NOAA, as well as ongoing governance • Lack of control over NOAA budget/schedule
FAA/Vendor	• High-reliability network in place for legacy products • Leveraging of existing FAA systems • Leverage private sector R&D and transition to operations capabilities • Control of budget and schedule	• Single weather vendor lock-in for a subset of weather products • Possible restrictions on the use of data introduced • Possibly redundant weather processing data centers (maintenance issue)

From a purely technical viewpoint, all of these alternatives are feasible. The key technical risk appears to be the reliability and latency of networks outside the FAA realm. This is currently somewhat of an unknown and will need to be further characterized to address it in a quantitative sense. From an FAA weather community perspective, another key risk may be a perceived lack of control over alternatives that include NOAA or a vendor. This perception of this risk would be reduced if a solid governance framework that provides the FAA with some flexibility and independence is created. Other concerns, such as lack of control over NOAA's budget and schedule, lie outside the scope of this study.

6.3 FAA IN-HOUSE ALTERNATIVES

The FAA in-house alternative consists of four subalternatives. Drawing upon the material in Section 4, Table 6 below presents some of the pros and cons associated with each.

TABLE 6

Pros and Cons of FAA In-House Alternatives

	Pros	Cons
WARP as NWP Baseline	• Highly reliable operational system • Architecturally aligned with NextGen layered approach. Processing generally separated from I/O • Combination of distributed and centralized processing • Modern processor (64 bit), modular software • SMP programming model • Recent WARP tech refresh started to align I/O with NNEW/SWIM I/O technologies • New hardware at ARTCCs (WINS box) could likely be leveraged to run NNEW data distribution reference implementations • Could be leveraged as I/O aggregation/dissemination node	• RAMP hardware still needed to drive DSR displays (processing not completely separated from I/O) • Processing not scalable to match future NWP requirements (e.g., CoSPA)
ITWS as NWP Baseline	• Highly reliable operational system • Distributed processing model • Modern processor (32 bit) • SMP programming model • Natural data aggregation node (TDWR radars) • SWIM interfaces in development. Partial reuse possible • Adhering to NNEW data formats. Partial reuse possible	• Proprietary operating system overlay (NOS) requires expertise to use effectively • Processing not scalable to match future NWP processing requirements (e.g., CoSPA) • Not yet adopting NNEW 4-D cube interfaces
CIWS as NWP Baseline	• Modern processing cluster (64 bit) • Centralized processing model • Highly scalable to match future NWP processing requirements • Hybrid SMP/Cluster programming model (though there is no standard for this yet)	• Nonoperational prototype system • No automatic fault recovery to protect against countrywide outages. Processor faults, although detected automatically, require human intervention to fix (no automatic fail-over)

	Pros	Cons
	- NNEW 4-D cube data formats/interfaces in development. Reuse possible	- Current NEXRAD ingest is prone to failure by region; however, fault tolerant ingest will be available in 2010
New System	- Avoid rework – get design and implementation right (in the NextGen ballpark) the first time	- Likely to take longer to get initial version of NWP prototype up and running using a from-the-ground-up approach

During this evaluation, we found that these alternatives, based on one system being considered “the baseline,” were difficult to distinguish. From an engineering perspective, each system has its strengths and weaknesses, and the sense is that the strengths of each will be incorporated into the final system, regardless of which of the original systems is considered to be the baseline. Any other path implies extra effort in terms of throwaway work that must later be corrected, or a suboptimal final implementation. Using the WARP system baseline as an example, the natural course is to leverage the highly reliable NEXRAD ingest infrastructure and look to CIWS for highly scalable processing component. This outcome is true whether one starts with either WARP or CIWS as the baseline system. For this reason, we recommend the fourth option, which consists of a best-of-breed approach that we feel would naturally emerge in all four of these options as stated.

6.4 COMMENTARY ON ALTERNATIVES

Although the stated alternatives are focused on NWP Segment 1, by implication, they set the general direction for subsequent NWP segments as well. Care should be taken to ensure that the approach selected for segment 1 is consistent with the broader NextGen vision. This raises the following questions:

- Are the FAA subalternatives sufficiently different from one another? Each FAA weather system has strengths and weaknesses. As described in the previous section, an NWP implementation will naturally leverage the strengths of each to achieve the desired result. The benefit of considering one of the existing programs (WARP, ITWS, CIWS) as the baseline is unclear.
- The alternatives as stated can be interpreted as being quite rigid. They effectively rule out, for example, weather products being generated by a mix of external organizations (NWS and a commercial vendor) for NWP Segment 1. This is likely to have an effect on future segments as well. This seems counter to the NextGen vision of seamless information sharing and its associated agility benefit.
- The alternatives as presented focus on a model whereby weather *products* are produced by the FAA, the NWS, or a commercial vendor. There is little mention of processing infrastructure as

a separable concept. This approach has the potential to simply re-establish new weather system silos that will be difficult to change over time (though easier than before due to the use of shared infrastructure). Adding alternatives that focus on research and operational *processing resources* rather than only weather *products* would result in a more diverse set of alternatives. This would be in line with the architectural guidance provided in Section 4.

An example of a modified set of alternatives reflecting the above feedback is provided in Table 7. This is included to foster discussion rather than as a recommendation, since the focus of the study was the specified, approved set of alternatives.

TABLE 7

Incorporating Processing Infrastructure Alternatives

Weather *Products* as a Service	Alternative 1. FAA generates and provides aviation-specific weather products for Air Traffic Control (ATC) community
	Alternative 2. Combination of FAA and external organization(s) generate and provide aviation weather products for ATC community (external organizations include NWS and/or commercial vendors)
	Alternative 3. External organization(s) provide aviation-specific weather products for ATC community
Weather *Processing Infrastructure* as a Service	Alternative 4. FAA hosts processing infrastructure for aviation-specific weather products
	Alternative 5. Combination of FAA and external organization(s) host processing infrastructure for aviation-specific weather products
	Alternative 6. External organization(s) host processing infrastructure for aviation-specific weather products

7. SUMMARY AND RECOMMENDATIONS

The NWP is intended to replace the processing component of a number of existing, stove-piped weather systems with an agile, scalable processing infrastructure that leverages a number of other NextGen infrastructure programs. A number of alternatives paths have been proposed to achieve the first step in the transition, termed segment 1, which is scheduled for the 2013 time frame. This study has examined the alternatives from a technical perspective, resulting in the following findings and recommendations for future research.

7.1 FINDINGS

1. In a net-centric, service-oriented environment, there is significant flexibility with regard to the choice between distributed or more centralized processing solutions. In the presence of a reliable, cost-effective communications network, solutions are naturally driven towards a more centralized model for reasons of maintainability and ease of implementation of data-fusion algorithms.

2. In order to maintain the desirable composability property of a service-oriented architecture, careful attention should be paid to the granularity of processing components that conform to NNEW/SWIM service interfaces. As a general rule, the components should be made as fine-grained as is practical, within the constraints of algorithm efficiency.

3. Processing clusters based on modern multicore architectures present a hybrid hardware environment that combines a symmetric multiprocessor architecture with a “classic” cluster architecture. The programming models currently lag the hardware implementations, and still typically focus on one or the other. This results in a significant manual coding effort to optimally make use of the hardware, increasing cost and schedule. This is an active area of research in the high-performance computing community.

4. Fault tolerance in operating large processing clusters is an area of concern. With large numbers of compute nodes, failures of a single node can be relatively common. We do not view that a primary/backup system is necessarily a complete solution. An approach that combines the primary/backup approach with a level of automated fault recover in each individual system would be preferable. Virtualization technologies are a strong candidate for use in this application. Licensing cost is potentially a significant issue if commercial virtualization solutions are to be considered.

5. Within the FAA domain, the FTI infrastructure should provide network connectivity with sufficient bandwidth and reliability to meet the requirements for NWP Segment 1. Replacement of the low-latency terminal products may be possible with existing FTI infrastructure, or may require modifications to FTI to classify and prioritize traffic (quality of service). In either case,

we view this as a very achievable goal for the segment 2 time frame. Ongoing research in the NNEW program should help to clarify this issue in the early FY 11 time frame.

6. NWP alternatives that cross the FAA organizational boundary (NWS, commercial vendor) come with an associated reliability risk since external networks are not necessarily designed to the same requirements as FTI. The bandwidth through the FTI ED-8 gateway is also a technical risk for high-volume weather products, though we view that as a problem that can be addressed by 2013 if allocated sufficient resources.

7. NWP alternatives that include a commercial vendor are subject to vendor lock-in. It is recommended that vendor-focused alternatives be required to support a clear upgrade path for weather product generation algorithms delivered by the aviation weather R&D community.

8. The alternatives as currently worded focus on legacy and advanced products rather than the product processing infrastructure. The difference is subtle but has important implications with respect to NWP multiagency agility. In other words, an NWS-generated product is a different thing than an NWS-hosted processing infrastructure capable of running algorithms provided by multiple agencies. The latter approach has proven its worth in the context of the NEXRAD OpenRPG and in the cloud computing community as well.

7.2 FOLLOW-ON RESEARCH

Recommendations for follow-on research are broken down into two categories. IT infrastructure research focuses on generic IT infrastructure and the interaction between NWP and the other NextGen infrastructure programs. Weather systems research focuses on the details of FAA weather systems and how to best transition from today’s weather systems to the long-term NextGen vision.

7.2.1 Information Technology Infrastructure Research

- **Survey of hybrid symmetric multiprocessor/message passing programming models.** This research would survey and evaluate current trends in programming models for multicore cluster architectures and provide recommendations for NWP in the near and long term. This is best accomplished as a collaborative effort between NWP stakeholders, with coupling to ongoing research in the high-performance computing community.

- **Assessment of virtualization technologies and their application to NWP.** This research would survey the different virtualization technologies available and provide recommendations on how to leverage the technologies in the NWP implementation. Focus areas for this research would include use of virtualization in the context of fault tolerance, as well as use as a generic deployment “container” for weather algorithms in the cloud-computing processing model.

- **Investigation of cloud-computing deployment models.** This research would investigate the state of the art with respect to cloud-computing deployment models, as well as future directions. The outcome of the research would be a set of recommendations on how the overall approach may be of utility to the NWP.

- **Verification of quality of service for low-latency applications.** In order to ensure that weather processing currently tightly coupled to a particular location (e.g., ITWS in the terminal area) is capable of becoming location-agnostic in the future, some requirements analysis and validation testing involving the NNEW/SWIM/FTI infrastructure will be required. This recommendation assumes that research ongoing in the NNEW/SWIM/FTI programs will provide the core QoS capability. The role of the NWP research would be to help drive QoS requirements and perform the necessary testing early on.

- **Assessment of network reliability and latency of external data-provider networks.** This research is needed to understand near- and longer-term risks associated with NWP alternatives that depend on non-FAA networks, particular in the case of weather products that are considered safety critical. This would obviously be a collaborative effort with the other NWP stakeholders.

7.2.2 NextGen Weather Capabilities Transition Research

- **Detailed analysis of current operational capabilities transition.** This study has provided some initial high-level information regarding the pros and cons of current NAS weather systems when viewed in the future NextGen context. Follow-on research is required to provide the additional detail to more thoroughly assess the implementation alternatives. Rather than initially focusing on particular systems, we recommend that the work start by looking broadly at algorithmic needs, decomposing the weather processing functionality into modular, composable blocks. This would be followed up by a more in-depth look at functionality in existing systems, resulting in recommendations of how to best leverage those systems to implement the modular functional blocks.

- **Assess opportunities for product improvements.** Though system consolidation and compatibility with NextGen infrastructure programs are worthy objectives in their own right, NWP Segment 1 should ideally demonstrate a number of improved capabilities for the end users. This research would focus on coming up with a set of candidate product improvements, including assessments of likely user acceptance and strategies for effectively and efficiently folding in the improvements into the NWP over time.

This page intentionally left blank.

GLOSSARY

ADDS	Aviation Digital Data Service
API	Application Programming Interface
ARTCC	Air Route Traffic Control Center
ASR-9	Airport Surveillance Radar-9
ATOP	Advanced Transport Operating System
AWOS	Automated Weather Observation System
BT	Briefing Terminal
CDDS	CIWS Data Distribution Service
CIWS	Corridor Integrated Weather System
CONUS	Continental United States
CoSPA	Consolidated Storm Prediction for Aviation
COTS	Commercial Off-The-Shelf
CPU	Central Processing Unit
DMA	Direct Memory Access
DoD	Department of Defense
DOTS	Dynamic Ocean Tracking System
DSR	Display System Replacement (Enroute)
ESB	Enterprise Service Bus
ERAM	En Route Automation Modernization
FAA	Federal Aviation Administration
FFT	Fast Fourier Transform
FPGAs	Field-Programmable Gate Arrays
FTI	Federal Telecommunications Infrastructure
GPU	Graphics Processing Unit
HTTP	Hypertext Transfer Protocol
I/O	Input/Output
IP	Internet Protocol
IT	Information Technology
ITWS	Integrated Terminal Weather System

JMS	Java Message Service
JMX	Java Management Extensions
LDM	Local Data Manager
LLWAS	Low Level Windshear Alert System
MIGFA	Machine Intelligent Gust Front Algorithm
MPAR	Multifunction Phased-Array Radar
MPI	Message Passing Interface
NAS	National Airspace System
NEXRAD	Next Generation Weather Radar
NFU	National Filter Unit
NNEW	NextGen Network-Enabled Weather
NOAA	National Oceanic and Atmospheric Administration
NOS	Network Operating System
NWP	NextGen Weather Processor
NWS	National Weather Service
OSGi	Open Systems Gateway Interconnect
PC	Personal Computer
QoS	Quality-of-Service
RAMP	Radar Acquisition and Mosaic Processor
SIPs	SWIM-Implementing Programs
SMP	Symmetric Multiprocessor
SOA	Service-Oriented Architecture
SOAP	Simple Object Access Protocol
SWIM	System Wide Information Management
TDWR	Terminal Doppler Weather Radar
TRACONS	Terminal Control Center
TWIP	Terminal Weather Information for Pilots
URET	User Request Evaluation Tool
WAN	Wide-Area Network
WARP	Weather and Radar Processor
WINS	Weather Information Network Server

WSDL	Web Services Description Language
WSP	Weather Systems Processor
XML	Extensible Markup Language
XMPP	Extensible Messaging and Presence Protocol

This page intentionally left blank.

REFERENCES

1. Heroux, M., “Some Thoughts On Multicore,” Sandia National Laboratories, http://www.cs.sandia.gov/~maherou/docs/SomeThoughtsOnMulticore.pdf.

2. Eadline, D., “Interconnects, 10GigE and Infiniband,” 8 July 2009, http://www.clusterconnection.com/2009/07/interconnects-10gige-and-infiniband.

3. Shainer, G., Liu, T., Michalakes, J., Liberman, J., Layton, J., Celebioglu, O., Schultz, S.A., Mora, J., and Cownie, D., “Weather Research and Forecast (WRF) Model Performance and Profiling Analysis on Advanced Multi-core HPC Clusters,” http://www.linuxclustersinstitute.org/conferences/archive/2009/PDF/Shainer_64557.pdf.

4. Stobie, J., Moosakhanian, A., Jackson, P., Brown, W., “Evolution of FAA's Weather and Radar Processor (WARP) into the Next Generation Air Transportation System,” Conference Proceedings, American Meteorological Society, 2008, http://ams.confex.com/ams/pdfpapers/132876.pdf.

5. Clark, David, “The Design Philosophy of the Internet Protocols,” Computer Communications Review, August 1988.