

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

TESLA, INC.

Petitioner,

v.

THE UNITED STATES OF AMERICA AS REPRESENTED BY THE
SECRETARY OF THE NAVY
(INTELLECTUAL VENTURES II LLC)

Patent Owner.

Case No. IPR2025-00341
U.S. Patent No. 7,181,743

PATENT OWNER'S PRELIMINARY RESPONSE

TABLE OF CONTENTS

PATENT OWNER’S EXHIBIT LIST	v
I. INTRODUCTION	1
II. SPECIFICATION OF THE ‘743 PATENT	3
III. CLAIMS OF ‘743 PATENT.....	7
IV. REFERENCES RELIED ON IN THE PETITION	8
A. Declaration of Erez Zadok (EX1003).....	8
B. Welch- “Distributed, Scalable, Dependable Real-Time Systems: Middleware Services and Applications” (EX1005).....	9
C. Ravindran - “Specification and Modeling of Dynamic, Distributed Real-time Systems.” (EX1006) (also EX2007)	13
D. Hosokawa - US6088727 (EX1007).....	14
V. THE PETITION FAILS TO DEMONSTRATE A REASONABLE LIKELIHOOD OF SUCCESS ON THE MERITS	14
A. Ground 1 Is Deficient Because Petitioner Fails to Show That Welch and Ravindran Disclose the Claimed Event Correlation, Application Health, QoS, or Configuration Functions.	16
1. Petitioner Fails to Show That Welch teaches “a second function group which provides application computer program ... event correlation capabilities” as recited in the challenged claims.	17
a) Petitioner Mischaracterizes Welch’s Radar-Based Tracking Algorithms as Application-Level Event Correlation Capabilities.....	19
b) Petitioner’s Reliance on Inter-Application Communication in Welch to Establish Event Correlation Capabilities Is Unsupported, Speculative, and Legally Insufficient	26
c) Petitioner Fails to Show That Welch Discloses the Claimed Event Correlation Capabilities, Relying Instead on Mischaracterized Disclosures and Unsupported Expert Testimony.....	33
2. Petitioner Fails to Show That Welch’s Disclosure of “Paths” Satisfies the Claimed Requirement of Determining the State and Health of Distinct “Application Programs.”	34

a)	Petitioner Fails to Establish That Welch’s “Paths” Correspond to the Claimed “Application Programs” Whose State and Health Must Be Determined	35
b)	Petitioner’s Expert Improperly Equates “Paths” with “Application Programs” Despite Welch’s Clear Distinction Between the Two	40
3.	Petitioner Fails to Show That Welch Discloses QoS Requirements Applied to Individual RTCS Application Programs, Rather Than System-Level Objectives.....	43
a)	Petitioner Fails to Show That Welch Applies QoS Requirements to RTCS Applications Rather Than to the System as a Whole	44
b)	Petitioner’s Reliance on Welch’s Resource Reallocation Disclosures Fails to Show That QoS Requirements Are Applied to Individual Application Programs	47
4.	Petitioner Fails to Show That Welch Discloses the Distinct Step of “Configuring” Application Programs, as Required by the Challenged Claims.....	49
a)	Petitioner Fails to Show That Welch Discloses “Configuring” Application Programs as Required by the ‘743 Patent.....	50
b)	The Petition Renders the Term “Configuring” Redundant, Contrary to Federal Circuit Precedent Requiring Distinct Claim Terms to Have Distinct Meanings.	53
c)	Petitioner Fails to Show That Welch Discloses the Distinct Step of “Configuring” Application Programs as Required by the Challenged Claims.	55
5.	The Suggested Addition of Ravindran does not cure the shortcomings of Welch.....	57
6.	Petitioner Fails to Establish Unpatentability Because Welch and Ravindran Do Not Disclose the Claimed Event Correlation, Application Health Determination, Application-Specific QoS, or Configuring Functions.....	58
B.	Ground 2 Is Deficient Because Hosokawa Does Not Cure the Shortcomings of Welch and Ravindran	60
VI.	SUMMARY OF ARGUMENTS ON THE MERITS	61

VII. CONCLUSION.....	64
----------------------	----

PATENT OWNER'S EXHIBIT LIST

EX2001	Petitioner's Answer to Complaint and Counterclaim, dated June 20, 2024, in <i>Intellectual Ventures II LLC v. Tesla, Inc.</i> , C.A. 1:24-cv-00884-ADA (W.D. Tex.)
EX2002	Docket from District Court Litigation, <i>Intellectual Ventures II LLC v. Tesla, Inc.</i> , C.A. 1:24-cv-00884-ADA (W.D. Tex.)
EX2003	Scheduling Order from District Court Litigation, <i>Intellectual Ventures II LLC v. Tesla, Inc.</i> , C.A. 1:24-cv-00884-ADA (W.D. Tex.)
EX2004	Defendant's Second Supplemental Preliminary Invalidity Contentions, dated January 9, 2025, in District Court Litigation, <i>Intellectual Ventures II LLC v. Tesla, Inc.</i> , C.A. 1:24-cv-00884-ADA (W.D. Tex.)
EX2005	Letter from Patent Owner to Associate General Counsel for Petitioner, dated August 20, 2020, enclosing copy of U.S. Pat. No. 7,181,143 and Claim Chart Presentation
EX2006	L. R. Welch, B. Ravindran, R. Harrison, L. Madden, M. Masters and W. Mills, "Challenges in Engineering Distributed Shipboard Control Systems," The IEEE Real-Time Systems Symposium, December 1996."
EX2007	L. R. Welch, B. Ravindran, B. Shirazi and C. Bruggeman, "Specification and Modeling of Dynamic, Distributed Real-Time Systems," in Proceedings of the 19th IEEE Real-Time Systems Symposium, 72-81, IEEE Computer Society Press, 1998.
EX2008	Declaration Of Dr. Sanjay Ranka In Support Of Patent Owner's Preliminary Response.
EX2009	Dr. Sanjay Ranka CV

I. INTRODUCTION

The Petition (Paper 2) challenging claims 10, 11, 14, 16, and 18 (the “Challenged Claims”) of US Patent 7,181,743 (the “’743 Patent) fundamentally mischaracterizes the teachings of the asserted prior art references, improperly relies on conclusory expert testimony, and fails to address critical limitations expressly recited in the claims. As a result, the Petition does not demonstrate a reasonable likelihood of success on the merits.

The ‘743 Patent discloses a dynamic, distributed resource management system that intelligently monitors the health of hosts, networks, and application programs, correlates raw event data into composite events for higher-level decision-making, dynamically configures and controls application programs, and manages system behavior to satisfy defined Quality-of-Service (QoS) objectives. Each element recited in the challenged claims—such as application-level event correlation, individual application health determination, application-specific QoS enforcement, and application configuration—is essential to the patented system’s ability to deliver real-time, adaptive management of distributed computing environments.

Petitioner’s reliance on Welch, Ravindran, and Hosokawa fails to meet its burden. Welch’s disclosures, properly understood in light of the HiPer-D Testbed and the dynamic path paradigm described in Ravindran, pertain to sensor output used

by radar-based systems to track radar targets like missiles, not application-level monitoring and management. Petitioner improperly equates radar target tracking with software event correlation, conflates paths with applications, and fails to show that QoS requirements are applied to individual applications. Petitioner likewise fails to demonstrate that Welch teaches the distinct act of configuring application programs, improperly collapsing distinct claim terms and rendering them superfluous.

Further, Ravindran is cited only for system-level hardware monitoring, not for addressing any of the missing claim elements. Hosokawa, cited in Ground 2, similarly fails to cure the deficiencies, as it relates only to unrelated aspects of software storage and package management.

Compounding these deficiencies, Petitioner relies heavily on an expert declaration that merely parrots the Petition's arguments without offering meaningful technical analysis or independent reasoning. Such conclusory testimony is entitled to little, if any, evidentiary weight.

Accordingly, the Petition does not establish a reasonable likelihood that Petitioner would prevail with respect to any challenged claim. Institution should be denied.

II. SPECIFICATION OF THE ‘743 PATENT

The ‘743 Patent discloses a system for dynamic resource management in distributed computing environments. The invention focuses on addressing the challenges associated with managing large-scale, interconnected computer systems running multiple applications, where resource availability and operational conditions can change unpredictably. EX1001, 2:20-32; EX2008, ¶28.

The Patent describes a distributed resource management architecture that continuously monitors both hardware resources and application programs to ensure that Quality-of-Service (QoS) objectives are maintained despite failures, load spikes, or environmental changes. EX1001, 4:8-16. The system is designed to operate dynamically, adapting in real time to maintain dependability, scalability, and survivability across distributed nodes. EX1001, 4:25-29; EX2008, ¶31-32.

The architecture includes several important functional components working together to achieve these goals. Host Monitors collect system-level metrics such as CPU usage, memory consumption, and network traffic to provide visibility into the health of hardware resources. EX1001, 10:15-29. Instrumentation Daemons provide application-level monitoring by collecting internal application-level status information, error conditions, and performance metrics directly from running applications. EX1001, 12:-64-13:2 This raw data is then passed to Instrumentation Correlators, which transform it into higher-level composite events based on user-

defined grammars interpreted at runtime. EX1001, 13:9-15; EX2008, 34-38. The event correlation process identifies meaningful patterns and conditions across distributed applications, such as combining individual performance warnings into a composite event. EX1001, 50:56-63; EX2008, 41. For example, instead of merely logging that Application A emitted “low level event X” (e.g., an application overload) and Application B emitted “low level event Y” (an application error), the Instrumentation Correlators recognize that when these events occur within a certain time window or frequency, they together signify a higher level composite event like a “data processing pipeline overload.” These composite events are then used by other components—such as the QoS Managers, or Resource Manager—to guide automated responses like scaling applications, or reallocating tasks. EX2008, ¶37

Data collected by the monitors and correlators is stored by a History Server, enabling retrospective analysis and trending, which informs future decision-making. EX1001, 44:41-47; EX2008, ¶34. The Resource Manager is the central decision-making component that analyzes incoming real-time and historical data to determine necessary resource adjustments. EX1001, 51:52-53. It can issue commands to start, stop, move, or replicate applications across hosts to optimize system behavior and meet QoS targets. EX1001, 51:53-52:6. Decisions made by the Resource Manager are executed by Program Control Agents, which act to reallocate applications based

on resource availability, application criticality, and overall system goals. EX1001, 24:5-17; EX2008, ¶38.

The ‘743 Patent highlights the importance of event correlation as a distinguishing feature of the invention. Rather than merely recording that discrete events have occurred in isolation; the system analyzes these events to infer higher-level conditions that require intervention. EX1001, 42:44-49. As explained by Dr. Ranka, “this capability enables the system to move beyond passive monitoring toward active management based on intelligent interpretation of distributed system behavior.” EX2008, ¶46.

A critical enabling feature of the resource manager is its reliance on a System Specification File, which defines the hardware architecture, application structure, and QoS requirements. EX1001, 16:19-45, EX2008, ¶36. Using this specification, the Resource Manager makes dynamic decisions about where applications should be run, how they should be configured, and when corrective actions must be taken to prevent service degradation. EX1001, 18:31-36, EX2008, ¶39.

In the ‘743 Patent, the system’s ability to dynamically configure applications is specifically emphasized. EX2008, ¶39. By way of example, configuring an application involves setting runtime parameters, such as environment variables, working directories, executable paths, and command-line arguments, which may include dynamically resolved information like hostnames or user identifiers.

EX1001, 27:23-28:60. As explained by Dr. Ranka, “[p]roper configuration necessary to ensure that applications operate correctly within the distributed environment and can appropriately interact with dependent applications and resources.” EX2008, ¶39.

Accordingly, the overall objective of the ‘743 Patent is to provide a framework for real-time, adaptive control over distributed computing resources. By integrating hardware and software monitoring, application-level event correlation, dynamic resource allocation, and configuration management, the disclosed embodiments ensure that distributed computing environments can maintain high dependability, timeliness, and availability even under unpredictable and challenging conditions. EX2008, ¶47-48. This dynamic QoS-driven approach overcomes limitations in the prior art and allows mission-critical systems to respond intelligently to failures, bottlenecks, and workload changes without manual intervention. EX1001, 2:15-19, EX2008, ¶48.

In summary, the ‘743 Patent discloses a distributed system that individually monitors the health of hosts, networks, and application programs, transforms raw event data into composite events for intelligent decision-making, dynamically configures and controls application programs, and actively manages system behavior in real time to satisfy defined Quality-of-Service (QoS) objectives. As explained by Dr. Ranka, “these capabilities represent significant improvements over prior art

static resource management systems, positioning the invention as a comprehensive solution for survivable, scalable, and self-managing distributed computing architectures.” EX2008, ¶49.

III. CLAIMS OF ‘743 PATENT

Claim 10 of the ‘743 Patent is only the independent claim among the challenged claims. Claim 10 recites

10. Software stored on at least one host for converting N networked hosts into a resource managed system instantiating M managed characteristic application computer programs, each managed characteristic application computer program managed by one of the N networked hosts, the software comprising:

a first function group which monitors the host and network resources;

a second function group which provides application computer program event reporting and event correlation capabilities;

a third function group which provides reasoning and decision making capabilities for the resource managed system, wherein the third function group comprises:

a first function which determines a state and health of the N hosts, a network operatively coupling the N hosts to one another and the M managed characteristic application computer programs in the distributed environment;

a second function which determines required allocation and reallocation actions needed to maintain a plurality of Quality of Service (QoS) requirements established for the M managed characteristic application computer programs, the QS requirements dictating parameters regarding service quality of the M management characteristic application programs; and

a third function which generates automatic control signal requests corresponding to the actions dictated by the QoS requirements, such that the managed characteristic application computer programs are moved, shutdown, and started in accordance with satisfaction of the QoS requirements; and

a fourth function group which provides program control capabilities permitting starting, stopping, and configuring of selected ones of the M managed characteristic application computer programs on respective ones of the N hosts in the resource managed system,

where M and N are positive integers and where M may be equal to, greater than, or less than N.

EX1001, 56:46-57:18.

IV. REFERENCES RELIED ON IN THE PETITION

A. Declaration of Erez Zadok (EX1003)

Petitioner's Expert Declaration (EX1003) from Dr. Zadok is not a piece of prior art, but it nonetheless is being used in the Petition to fill gaps in the prior art. Dr. Zadok's Expert Declaration should be afforded little, if any, evidentiary weight. As an initial matter, large portions of the declaration simply mirror the Petition with only minimal, insubstantial differences. For example, paragraphs 80–88, 96–98, 104–117, 122–139, 141–146, 148–149, 151–163, 165–173, 175–191, 193–195, 197–201, 203–210, 214, 216–220, 222–224, 226–227, 229–232, 234–235, 237–244, 246–249, 251–255, 258–260, 262, 264–265, 267–270, 273–282, 291–292, 294–300, 302–305, and 308–316 of the expert declaration essentially parrot the Petition word-for-word. The few variations between the Petition and the expert declaration are largely superficial—for instance, in many instances where the Petition provides a citation to a reference, the declaration merely adds a direct quotation without offering any meaningful analysis, technical insight, or independent reasoning.

Such rote repetition is entitled to little evidentiary weight. As the USPTO has explained in *Xerox Corp. v. Bytemark, Inc.*, IPR2022-00624, Paper 9 at 15–17 (PTAB Aug. 24, 2022) (precedential), expert testimony that merely restates the Petition’s conclusory assertions without offering additional substantive explanation is entitled to “little weight.” The Federal Circuit has similarly affirmed that an expert declaration adds no meaningful value where it simply repeats a petitioner’s arguments without independent analysis. See *Facebook, Inc. v. Windy City Innovations, LLC*, 973 F.3d 1321, 1340–41 (Fed. Cir. 2020) (affirming the Board’s finding that expert testimony was unpersuasive where it “merely repeated Petitioner’s argument, nearly verbatim, without citation to the basis for his testimony”).

Here, the Petition’s heavy reliance on an expert declaration that mostly parrots the Petition’s arguments underscores the weakness of Petitioner’s case and supports affording the declaration little weight.

B. Welch- “Distributed, Scalable, Dependable Real-Time Systems: Middleware Services and Applications” (EX1005)

Welch (EX1005), titled “Distributed, Scalable, Dependable Real-Time Systems: Middleware Services and Applications,” focuses on path-based resource management, where the system monitors the health of end-to-end control paths that span multiple distributed software components, networks, sensors, and actuators. When a path’s performance degrades, such as by missing deadlines, the system

responds by reallocating resources, replicating applications, or moving application programs to maintain overall path-level QoS. EX1005, 1.

Welch further describes experimental validation conducted using a Navy High Performance Computing (HiPer-D) testbed as described in “Challenges in Engineering Distributed Shipboard Control Systems.” EX1005, 2. “Challenges in Engineering Distributed Shipboard Control Systems” (EX2006) describes the development of the HiPer-D Program, a collaborative effort between DARPA and the US Navy. Within the HiPer-D Testbed, the authors implemented a distributed version of an anti-air warfare (AAW) system capable of tracking thousands of targets and managing multiple simultaneous engagements.

Key capabilities of the HiPer-D testbed described in “Challenges in Engineering Distributed Shipboard Control Systems” include managing weapon targeting and tactical display subsystems for operator interaction. The system is described to include Track correlation and filtering components that are responsible for processing radar sensor data, performing radar tracking initialization and refinement, and distributing radar tracking information to other modules via the Radar Track Data Server (RTDS). EX2006, 2 EX2008, ¶79-81.

Welch’s reference to HiPer-D testbed means that Welch’s middleware service was tested to track missiles for managing weapons targeting. These capabilities focus on processing real-time radar sensor outputs used by naval anti-air-warfare

operations to, e.g., determine a threat and guide a missile to destroy the threat once detected, and are not concerned with monitoring or managing individual application programs based on, e.g., application-level status information, error conditions, and performance metrics. EX2008, ¶82. In fact, Welch is silent with respect to how to monitor the radar tracking application itself. For example, the reference does not discuss monitoring the radar tracking application itself for an application overload or an application error condition.

Welch further cites to “Specification and analysis of dynamic, distributed real-time systems,” in Proceedings of the 19th IEEE Real-Time Systems Symposium, 72-81, IEEE Computer Society Press, 1998.” EX1005, 5. As shown in EX2007, the Paper that was presented at the conference was entitled “Specification and **Modeling** of Dynamic, Distributed Real-Time Systems.” Dr. Ranka and Petitioner agree that citation in Welch is a typographical error, and the proper citation is to Ravindran EX1006. See Petition, 39; EX2008, ¶60.

As noted by the Petition, Welch relies upon Ravindran to explain that his “approach is based on the dynamic path paradigm” and “[a] path-based real-time subsystem (see [Ravindran]) typically consists of a detection & assessment path, an action initiation path and an action guidance path.” Petition, 38; EX1005, 1. Specifically, Ravindran explains that a “subsystem can be modeled using three dynamic paths: threat detection, engagement, and missile guidance.” EX1006, 2.

Ravindran further explains that “[t]he threat detection path examines radar sensor data (radar tracks) and detects potential threats” and “the engagement path is activated, resulting in the firing of a missile to engage the threat.” Finally, Ravindran explains that “[t]he missile guidance path involves sensor hardware, software for filtering, software for evaluating & deciding, software for acting, and actuator hardware.” EX1006, 2. Accordingly, “the dynamic path paradigm described by Ravindran and cited to by Welch is the paradigm that analyzes radar data to determine a threat and guides a missile to destroy the threat once detected.” EX2008, ¶62.

This understanding of Ravindran’s dynamic path paradigm is consistent with Welch, which explains that in the dynamic path paradigm, “[t]he paths interact with the environment via evaluating streams of data from sensors, and by causing actuators to respond (in a timely manner) to events detected during evaluation of sensor data streams.” EX1005, 1. Therefore, “when read in the context of Ravindran, Welch clearly distinguishes the environmental data streams collected from radar sensors from the paths that perform real-time military functions such as threat detection, engagement, and missile guidance.” EX2008, ¶62.

As a result, Welch, when properly understood in light of its cited references to the HiPer-D testbed and Ravindran (EX1006), discloses a middleware system focused on managing end-to-end control paths in distributed, real-time military

environments, particularly for missile tracking and engagement operations. Its reliance on the HiPer-D testbed and the dynamic path paradigm described in Ravindran confirms that Welch's focus is on processing environmental sensor data for real-time military functions, including threat detection, missile engagement, and missile guidance. EX2008, ¶63.

C. Ravindran - "Specification and Modeling of Dynamic, Distributed Real-time Systems." (EX1006) (also EX2007)

Ravindran (EX2007), titled "Specification and Modeling of Dynamic, Distributed Real-Time Systems," introduces a programming-language-independent specification language designed to describe timing constraints and environment-dependent behaviors at the level of end-to-end "paths" through a distributed system. These paths span multiple applications, network elements, and hardware components, allowing system designers to model real-time constraints across complex, multi-component flows rather than focusing solely on individual applications. EX2008, ¶68.

Central to Ravindran's framework is the dynamic real-time path paradigm, which models real-time systems as consisting of multiple types of paths: situation assessment paths (e.g., continuous threat detection from sensor data), action initiation paths (e.g., launching a missile after detecting a threat), and action guidance paths (e.g., tracking and guiding a missile in flight). These paths interact

with the external environment by processing data streams from sensors and issuing control commands to actuators. EX2008, ¶69.

D. Hosokawa - US6088727 (EX1007)

Hosokawa discloses a cluster control system designed to monitor and manage resources across multiple computers in a cluster. Hosokawa's system enables the automatic transfer of application and service programs ("packages") from one computer to another when faults or failures occur, minimizing downtime. The system improves over prior art by allowing rapid and reliable failover, supporting prioritized and coordinated control of package execution, and maintaining system operations even when the manager itself encounters failures. EX2008, ¶71-75.

V. THE PETITION FAILS TO DEMONSTRATE A REASONABLE LIKELIHOOD OF SUCCESS ON THE MERITS

In Ground 1, Petitioner alleges that the challenged claims are unpatentable under 35 U.S.C. § 103 based on a proposed combination of Welch (EX1005) and Ravindran (EX1006). In Ground 2, Petitioner further alleges that the challenged claims are unpatentable under 35 U.S.C. § 103 based on a proposed combination of Welch (EX1005), Ravindran (EX1006), and Hosokawa (EX1007). However, neither ground establishes that all limitations of the challenged claims are taught or suggested by the cited references, individually or in combination.

As explained below, the Petition is deficient because it relies on mischaracterizations of the prior art, conclusory and unsupported expert assertions,

and improper speculation. First, the Petition fails to demonstrate that Welch discloses a second function group providing application-level event correlation capabilities, as required by the challenged claims. Petitioner's reliance on radar-based missile tracking algorithms and inter-application communication is misplaced and insufficient. Second, Petitioner does not show that Welch determines the state and health of distinct application programs; instead, Welch discloses monitoring complex paths that span multiple components. Third, the Petition fails to establish that Welch applies QoS requirements to individual applications, rather than merely articulating system-wide objectives. Fourth, the Petition fails to demonstrate that Welch discloses the distinct step of configuring applications, improperly conflating configuration with starting or stopping applications. Petitioner's further reliance on Ravindran does not cure these deficiencies, as Ravindran merely discloses system-level hardware load monitoring and does not address any of the missing claim limitations.

Finally, the addition of Hosokawa in Ground 2 likewise fails to remedy the shortcomings of Welch and Ravindran. Hosokawa is cited only for limited, unrelated teachings regarding software storage and numerical relationships between M and N; Hosokawa does not disclose, suggest, or cure the critical missing elements required by the challenged claims.

Accordingly, Petitioner has failed to meet its burden of demonstrating a reasonable likelihood of success under either Ground 1 or Ground 2, and institution should be denied.

A. Ground 1 Is Deficient Because Petitioner Fails to Show That Welch and Ravindran Disclose the Claimed Event Correlation, Application Health, QoS, or Configuration Functions.

The Petition, relying primarily on Welch and supplemented by Ravindran, fails to demonstrate that all limitations of the ‘743 Patent claims are disclosed or suggested. Petitioner mischaracterizes the prior art and depends on speculation and conclusory expert assertions that do not bridge critical gaps. Welch’s radar-based missile tracking algorithms are unrelated to application-level event correlation, and Petitioner’s alternative theory based on inter-application communication is speculative and unsupported. Welch also does not determine the state and health of distinct application programs, but instead monitors the health of complex “paths” spanning multiple components. Further, Welch applies Quality-of-Service (QoS) requirements only at the system level, not to individual Real-Time Control System (RTCS) applications, and fails to disclose the distinct act of configuring applications as separately required by the claims. Petitioner improperly conflates configuring with starting or stopping, contrary to Federal Circuit precedent, and fails to identify any technical setup processes required for configuration. Ravindran does not cure these deficiencies, as it addresses only hardware monitoring and omits any

disclosure of application-level event correlation, application health monitoring, application-specific QoS enforcement, or configuration.

Accordingly, Petitioner has failed to establish unpatentability under Ground 1 for at least the following reasons.

1. Petitioner Fails to Show That Welch teaches “a second function group which provides application computer program ... event correlation capabilities” as recited in the challenged claims.

The challenged claims of the ‘743 Patent recite, among other elements, “event correlation capabilities” provided by a second function group responsible for application-level instrumentation. As detailed in the specification, the “event correlation capabilities” are for transforming raw performance and status data into higher-level composite events that drive intelligent system behavior. EX2008, ¶¶85-87.

For example, the ‘743 Patent states:

The Instrumentation Correlators provide grammar-driven capabilities for correlating, combining, and reformatting application data into higher-level metrics (composite events) for use by displays or other Resource Management components. Each Correlator reads in a user-specified correlation grammar file that is interpreted at run-time by the Correlators instrumentation correlation engine. EX1001, 50:56-63

Accordingly, “in the context of the ‘743 Patent, ‘correlation capabilities’ refer to the system’s ability to analyze and interpret raw data from distributed applications in order to generate higher-level, meaningful insights—referred to in the patent as

composite events.” EX2008, ¶84. In the ‘743 Patent, composite events are produced by the Instrumentation Correlators (FG26A–FG26N), which use a grammar-driven engine to interpret sequences or patterns of application-generated data. EX1001, 14:34-41. These raw data points may include status messages, performance metrics, internal application states, or error reports. EX1001, 7:42-45. The correlators process this data based on user-defined rules to recognize when a meaningful condition has occurred. EX1001, 7:51-54. EX2008, ¶88.

Therefore, in the ‘743 Patent the “correlation process goes beyond basic communication or message passing; it actively derives context and meaning from distributed events, enabling more informed decisions by other system components.” EX20008, ¶87. For example, instead of merely logging that Application A emitted “low level event X” and Application B emitted “low eleven event Y,” the system may recognize that when these events occur within a certain time window or frequency, they together signify a higher level composite event like “data processing pipeline overload.” These composite events are then used by other components—such as the QoS Managers, or Resource Manager—to guide automated responses like scaling applications, or reallocating tasks. EX2008, ¶90.

Petitioner’s contention that Welch discloses or suggests the ‘743 Patent’s “event correlation capabilities” rests on a mischaracterization of Welch’s teachings and an unsupported extension of the reference’s actual disclosures. In particular,

Petitioner relies on Welch’s description of “track correlation and filtering algorithms” as allegedly demonstrating the claimed application level event correlation functionality, despite the fact that these algorithms are directed to radar-based missile tracking within a naval anti-air-warfare (AAW) system, as confirmed by “Challenges in Engineering Distributed Shipboard Control Systems” (EX2006), and are not concerned with monitoring or interpreting application-level software data. Petitioner’s expert further attempts to rely on Welch’s mention of inter-application communication to imply the presence of event correlation capabilities, but this theory conflates basic communication between applications with the distinct technical function of correlating application-generated events. Petitioner’s invocation of inherency to bridge this gap similarly falls short, as it fails to meet the high standard requiring that the claimed functionality necessarily be present in the prior art.

As a result, as explained in more detail below, Petitioner’s position lacks adequate support in the record and cannot establish that Welch teaches or suggests the claimed event correlation capabilities.

a) Petitioner Mischaracterizes Welch’s Radar-Based Tracking Algorithms as Application-Level Event Correlation Capabilities

Petitioner has failed to demonstrate that Welch discloses or suggests “a second function group which provides application computer program event reporting and

event correlation capabilities” as required by the challenged claims. The Petitioner’s reliance on Welch’s disclosure of “track correlation and filtering algorithms” is misplaced, as those algorithms pertain to radar-based missile tracking within an anti-air-warfare (AAW) control system—capabilities fundamentally unrelated to monitoring the application-level performance or status of software. Petitioner’s expert improperly attempts to bridge this gap by equating target tracking with application event correlation, thereby misconstruing the nature of the cited functionality.

As confirmed by the “Challenges in Engineering Distributed Shipboard Control Systems,”(EX2006) cited in Welch as Reference [10], Welch’s algorithms are directed to sensor data processing, not the collection and analysis of application program events such as application statuses, states, or internal errors. EX2008, ¶98. Accordingly, Petitioner’s arguments fail to meet its burden of demonstrating a reasonable likelihood that the challenged claims are unpatentable based on Welch, and institution should be denied.

Specifically, the Petition states, “Welch’s metrics calculation component (part of the “second function group”) provides “event correlation capabilities” because the QoS management software “implementation includes the following capabilities:...(2) track correlation and filtering algorithms.” Petition, 44-45. To

support this position, Petitioner sites its expert report (EX1003) at Par. [0158] which states:

158. Next, Welch’s metrics calculation component (part of the “second function group”) provides “event correlation capabilities” because the QoS management software “implementation includes the following capabilities: ... (2) track correlation and filtering algorithms.” EX1005, Welch, 4. For example:

The technology described in this paper was evaluated within the Naval Surface Warfare Center High Performance Computing (NSWC HiPer D) Testbed, which contains the experimental Navy system described in [10]. The implementation includes the following capabilities: (1) a simulated track source, **(2) track correlation and filtering algorithms**, (3) track data distribution services, (4) a doctrine server and three types of doctrine processing, (5) an engagement server, (6) a display subsystem including Xwindows based tactical displays, submode mediation, and alert routing surface operations, (7) a simulated weapons control system, and (8) identification upgrade capabilities. EX1005, Welch, 4.

EX1003, ¶158 (emphasis in original).

However, Petitioner’s expert appears to have failed to consider the first sentence of the block quote from Welch states that “The technology described in this paper was evaluated within the Naval Surface Warfare Center High Performance Computing (NSWC HiPer D) Testbed, which contains the experimental Navy system described in [10].” EX1005, 4. Reference [10] of Welch is “L. R. Welch, B. Ravindran, R. Harrison, L. Madden, M. Masters and W. Mills, “Challenges in

Engineering Distributed Shipboard Control Systems,” The IEEE Real-Time Systems Symposium, December 1996.” EX1005, 6.

When read in the context of “Challenges in Engineering Distributed Shipboard Control Systems” (EX2006), the “(2) track correlation and filtering algorithms” are for tracking missiles in an anti-air-warfare (AAW) control system and have absolutely nothing to do with “collecting and correlating application-provided data such as application statuses, states, performance, and internally detected errors.” EX1001, 7:43-45, EX2008, ¶94.

For example, “Challenges in Engineering Distributed Shipboard Control Systems” describes the development and implementation of a distributed anti-air warfare (AAW) control system within the HiPer-D (High Performance Distributed Computing) program, a joint effort by DARPA and the US Navy. Faced with the limitations of legacy military computing systems, HiPer-D explores scalable, fault-tolerant architectures using commercial off-the-shelf (COTS) hardware to support real-time operations across heterogeneous networks. The paper details the system’s modular structure, including sensors, decision-making algorithms, engagement management, and operator interfaces, all integrated within a distributed platform capable of tracking thousands of targets and handling multiple engagements. EX2008, ¶95-96.

Specifically, as shown in the following table, “Challenges in Engineering Distributed Shipboard Control Systems” recites virtually the same language as quoted from Welch.

“Challenges in Engineering Distributed Shipboard Control Systems” (EX2006)	Welch (EX1005)
The Testbed implementation of the AAW system includes the following capabilities: a simulated track source; <u>track correlation and filtering algorithms;</u> track data distribution services; a doctrine server and three types of doctrine processing (semi-auto, auto-sm, and auto-special); an engagement server; a display subsystem including X-windows based tactical displays, submode mediation, and alert routing; surface operations (reengineered from CMS-2 to Ada), and simulated weapons control system and identification upgrade capabilities. EX2006, 2	The implementation includes the following capabilities: (1) a simulated track source, <u>(2) track correlation and filtering algorithms,</u> (3) track data distribution services, (4) a doctrine server and three types of doctrine processing, (5) an engagement server, (6) a display subsystem including Xwindows based tactical displays, submode mediation, and alert routing surface operations, (7) a simulated weapons control system, and (8) identification upgrade capabilities.

EX2008, ¶97

Accordingly, when Welch references “track correlation and filtering algorithms,” Welch is referring to the capabilities of the anti-air-warfare (AAW) control system described in “Challenges in Engineering Distributed Shipboard Control Systems.” EX2008, ¶98.

“Challenges in Engineering Distributed Shipboard Control Systems” explains that:

Track Correlation and Filter (TCF) - TCF takes input from a sensor simulator and performs track initialization, radar track filtering, and related activities. The TCF component provides track data to the Radar Track Data Server (RTDS) for distribution to all other components of the system that require it.

Radar Track Data Server (RTDS) - Clients of the RTDS register for distribution of track data, specifying a minimum periodicity and a maximum tolerable latency. The RTDS then distributes the data received from the TCF function to the various clients at their specified rates. EX2006, 3

As a result, as shown in “Challenges in Engineering Distributed Shipboard Control Systems” Figure 1 (reproduced below), Track Correlation and Filter (TCF) algorithms are correlating and filtering the radar data and have absolutely nothing to do with monitoring how the AAW Software System is operating. EX2008, ¶97.

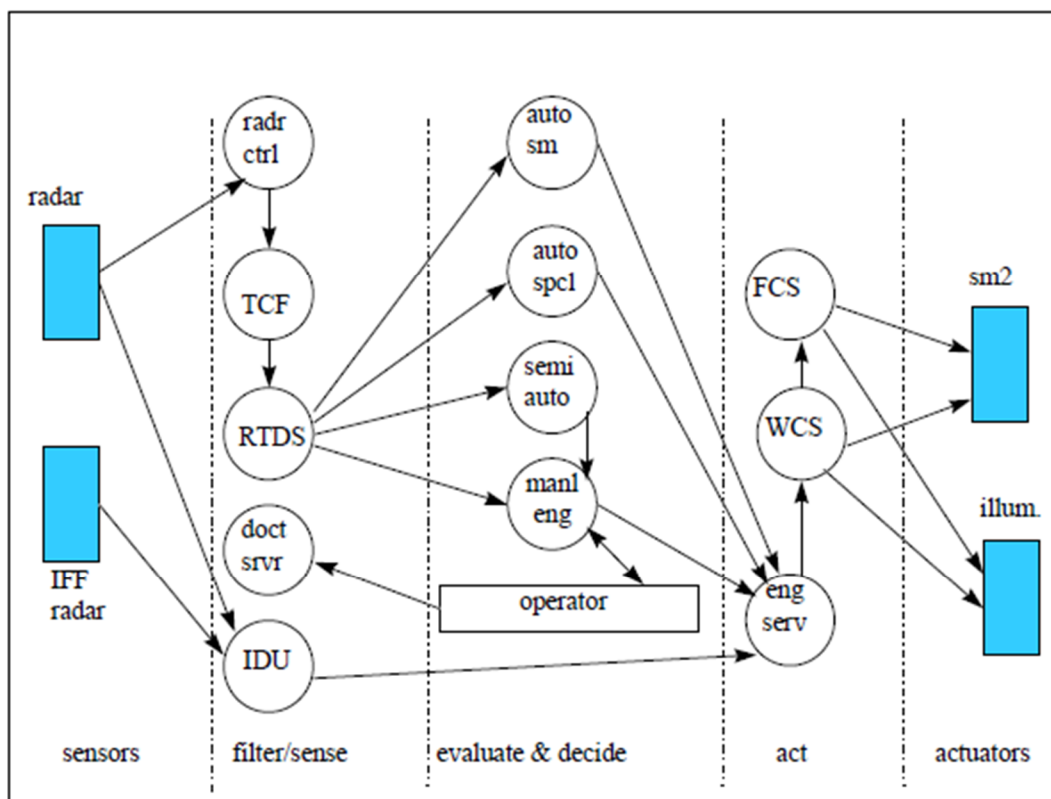


Figure 1. Elements of a shipboard AAW system.

EX2003, 3.

As a result, Petitioner’s expert statement that “In sum, Welch discloses “a second function group [e.g., software monitors] which provides application computer program [e.g., RTCS applications] event reporting [e.g., unhealth sub-paths, trend of load, etc.] and event correlation [e.g., correlation and filtering algorithms] capabilities,” (EX1003, ¶164) is clearly wrong. Petitioner’s expert has confused correlating and filtering radar data to track missiles with tracking the operations of a software system. EX2008, ¶98.

Therefore, Petitioner’s reliance on Welch to satisfy the limitation of “a second function group which provides application computer program event reporting and

event correlation capabilities” is fundamentally flawed. The Petition mischaracterizes Welch’s disclosure by equating radar-based tracking algorithms used in an anti-air-warfare (AAW) system with software mechanisms for reporting and correlating application-level events such as statuses, performance metrics, and internal errors. As demonstrated, Welch’s reference to “track correlation and filtering algorithms” pertains exclusively to missile tracking functionality evaluated in the context of the HiPer-D Testbed—a military control system described in detail in the cited paper “Challenges in Engineering Distributed Shipboard Control Systems.” That paper confirms that the cited algorithms are used to process radar data, not to monitor or correlate application-level software events.

b) Petitioner’s Reliance on Inter-Application Communication in Welch to Establish Event Correlation Capabilities Is Unsupported, Speculative, and Legally Insufficient

Petitioner contends that Welch inherently discloses the claimed “event correlation capabilities” based solely on its statement that applications in the system are “dependent (communicate with each other).” Relying on this passage, Petitioner’s expert concludes that such inter-application communication would be understood by a person of ordinary skill in the art (POSITA) to imply the presence of event correlation functionality. However, this argument rests on a flawed premise and a misreading of the cited disclosure.

Specifically, Petitioner's expert states, "given that Welch's "applications are dependent (communicate with each other)," a POSITA would have understood that the performance information of those applications includes "event correlation capabilities"—the ability to communicate messages and events between applications. EX1005, Welch, 2." EX1003, ¶159. The portion of Welch the expert appears to be relying upon in Welch states:

A major difference between the traditional load balancing techniques [7] and dynamic QoS-based resource management services lies in the overall goals. While load balancing systems (see Load-Leveler [5], LSF [12], NQE [2], PBS [4], Globus [3], and Condor [1]) attempt to achieve system performance goals such as minimized response time or maximized throughput, dynamic QoS-based resource managers strive to meet the QoS requirements of each application they manage. Another major difference between these systems is their workload models. Traditional load balancing systems assume independent jobs with known resource requirements. **In a dynamic resource management system, the workload requirements of applications can vary, based on environmental conditions; additionally, applications are dependent (communicate with each other).** EX1005, 2-3

There is absolutely nothing in the above passage describing application event correlation. As explained by Dr. Ranka, "the mere fact that two applications can communicate with each other does not inherently mean the system that includes the applications has correlation capabilities." EX2008, ¶101.

As Dr. Ranka explains:

Imagine two friends—Alice and Bob—are having a phone conversation. They're clearly communicating with each other,

exchanging information in real time. However, just because they're talking doesn't mean there's someone on the line analyzing both sides of the conversation to detect patterns, infer causes of misunderstandings, or summarize key points. For that, you'd need a third party—say, a translator or a meeting assistant—who listens to both sides, compares their statements, identifies important events, and connects the dots between them. More importantly, the phone does not have the ability to detect that since both Alice and Bob coughed more than a predetermined number of times during the phone conversation that the area in which Alice and Bob are located is experiencing a COVID outbreak.

Similarly, in a distributed system, two applications might send messages back and forth as part of a control workflow, but that communication alone doesn't mean the system has "correlation capabilities." Correlation requires a separate mechanism that observes, collects, and analyzes events from multiple sources to identify relationships, causes, or trends. Without that layer, the system may facilitate communication but lacks the intelligence to interpret or correlate the events.

EX2008, ¶102-103.

Therefore, Petitioner's reliance on the fact that applications in Welch's system are "dependent (communicate with each other)" fundamentally misunderstands the distinction between communication and event correlation because the mere existence of communication between entities does not imply the existence of a correlation mechanism capable of correlating events occurring at each of the entities. As Dr. Ranka's analogy shows absent a third party—a translator, meeting assistant, or other observer—there is no entity that monitors both sides of the conversation to detect patterns, identify key events, or draw inferences, such as recognizing that

multiple coughs could indicate a COVID outbreak. The communication channel itself (the phone) merely enables information transfer; it does not interpret or correlate the content of the messages.

Similarly, in Welch, the fact that applications can exchange messages does not mean that Welch's system has the "event correlation capabilities" required by the '743 Patent. Communication among applications simply enables the transmission of information necessary for distributed processing tasks; it does not inherently provide for the correlation of multiple discrete events into higher-level composite events based on predetermined rules or grammars, as taught by the '743 Patent. True event correlation requires an additional system layer that collects event data from distributed sources, analyzes it according to defined criteria, and generates higher-order insights to drive system behavior. Welch discloses no such mechanism. Therefore, Petitioner's attempt to satisfy the "event correlation" limitations based solely on the existence of inter-application communication is unsupported and legally insufficient. Without a distinct event correlation component—analogue to the independent interpreter in the expert's analogy—Welch's system lacks the critical capabilities required by the challenged claims.

As a result, Petitioner's reliance on inter-application communication to establish the claimed "event correlation capabilities" is misplaced and unsupported by the record. Instead, "the mere fact that two applications exchange messages—

like two people having a conversation.” EX 2008, 104. Specifically, event correlation requires more than communication; it demands a distinct component that observes, collects, and processes events across multiple sources to identify patterns or causal relationships. Welch contains no disclosure of such functionality, and Petitioner has failed to show otherwise. EX 2008, ¶105.

Further, as the Federal Circuit explains, the use of inherency, “a doctrine originally rooted in anticipation, must be carefully circumscribed in the context of obviousness” and “[t]he mere fact that a certain thing may result from a given set of circumstances is not sufficient to establish inherency.” *Par Pharm., Inc. v. TWi Pharm., Inc.*, 773 F.3d 1186, 1194-95 (Fed. Cir. 2014) (internal markings and citations omitted). The Court has “emphasized that the limitation at issue necessarily must be present in order to be inherently disclosed by the reference.” *Southwire Co. v. Cerro Wire LLC*, 870 F.3d 1306, 1310-11 (Fed. Cir. 2017) (internal citation omitted). Accordingly, “[a] party must...meet a high standard in order to rely on inherency to establish the existence of a claim limitation in the prior art in an obviousness analysis.” *Millennium Pharms., Inc. v. Sandoz, Inc.*, 862 F.3d 1356, 1367 (Fed. Cir. 2017). Therefore, obviousness cannot be established by the mere fact that something may happen. Instead, to teach the high standard of inherency, the alleged feature must necessarily be present in the cited reference.

In the present case, Petitioner has not met the high burden required to establish that the claimed event reporting and event correlation capabilities are inherently disclosed in Welch. The only support offered is the uncorroborated testimony of Petitioner’s expert, who asserts that because two applications in Welch communicate with each other, the system must inherently “provide[] application computer program event reporting and event correlation capabilities,” as recited in the challenged claims. However, there is nothing in the record—beyond this conclusory assertion—that explains how such communication necessarily results in the claimed functionality.

The Federal Circuit has made clear, “conclusory statements and unspecific expert testimony do not qualify as substantial evidence that could support the Board’s conclusions regarding obviousness.” *TQ Delta, LLC v. Cisco Sys.*, 942 F.3d 1352, 1360–61 (Fed. Cir. 2019). Likewise, the USPTO has recognized that expert declarations “merely offer[ing] conclusory assertions without underlying factual support” are entitled to little, if any, weight. *Xerox Corp. v. Bytemark, Inc.*, IPR2022-00624, Paper 9 at 15–17 (PTAB Aug. 24, 2022) (precedential). Petitioner’s expert testimony here suffers from precisely these defects: it lacks factual foundation, fails to tie any specific disclosure in Welch to the claim limitations, and offers only speculative reasoning to bridge a critical gap in the asserted prior art.

As a result, Petitioner has not shown that Welch discloses or suggests “a second function group which provides application computer program event reporting and event correlation capabilities,” as required by the challenged claims. The expert’s assertion that such capabilities are inherently present simply because the applications described in Welch “communicate with each other” is unsupported by the cited disclosure. The cited portion of Welch merely notes that the applications “communicate with each other” within a dynamic resource management system, but says nothing about collecting, reporting, or correlating application-level events such as performance metrics, states, or internal errors.

There is nothing in the record beyond the expert’s bare assertion to support the conclusion that inter-application communication necessarily implies the existence of event correlation functionality. The expert’s opinion fails to explain how or why such communication would give rise to the specific capabilities recited in the claims. The use of inherency in this context is improper, as it relies on speculation rather than a clear showing that the claimed functionality must be present in the prior art system.

Because Petitioner’s position depends entirely on uncorroborated and conclusory expert testimony, without factual support from Welch or any other cited reference, it fails to demonstrate a reasonable likelihood of success.

c) Petitioner Fails to Show That Welch Discloses the Claimed Event Correlation Capabilities, Relying Instead on Mischaracterized Disclosures and Unsupported Expert Testimony

Petitioner has failed to demonstrate that Welch discloses or suggests “a second function group which provides ... event correlation capabilities,” as required by the challenged claims. The Petition erroneously relies on Welch’s reference to “track correlation and filtering algorithms” as evidence of such capabilities. However, as confirmed by “Challenges in Engineering Distributed Shipboard Control Systems” (EX2006), those algorithms relate exclusively to radar-based target tracking within a naval anti-air-warfare (AAW) system, not to any mechanism for collecting or correlating application-generated performance data, statuses, or internal errors. The cited algorithms in Welch are used to process and distribute sensor data concerning physical targets—not to monitor application-level software events. Petitioner’s expert improperly conflates these distinct technical domains.

Petitioner also attempts to satisfy the claim limitations by asserting that, because applications in Welch are described as “dependent (communicate with each other),” a person of ordinary skill in the art would understand the system to include event correlation capabilities. This assertion is fundamentally flawed. As Dr. Ranka explains through a helpful analogy, the fact that two applications exchange messages is akin to two individuals having a conversation—it reflects communication, not correlation. Welch contains no disclosure of correlation functionality, and the expert

fails to identify any component or functionality that performs this type of analysis. Just like two friends talking on the phone does not imply the phone system can detect a COVID outbreak, communication between components in Welch does not imply the presence of a correlation engine or event monitoring capability. EX2008, ¶106.

Finally, Petitioner’s reliance on inherency is legally insufficient. The Federal Circuit has repeatedly emphasized that inherency in the obviousness context must be narrowly applied and cannot be based on mere possibility or speculation. The expert’s unsubstantiated testimony, which lacks factual support or a citation to any specific disclosure in Welch, does not meet this standard. As recognized in *TQ Delta and Xerox v. Bytemark*, conclusory expert declarations that fail to explain how the prior art teaches the claimed limitation are entitled to little weight. Because Petitioner’s argument rests entirely on mischaracterized disclosures and unsupported expert opinion, it fails to satisfy the burden of proof.

2. Petitioner Fails to Show That Welch’s Disclosure of “Paths” Satisfies the Claimed Requirement of Determining the State and Health of Distinct “Application Programs.”

Petitioner contends that Welch discloses the determination of the “state and health” of application programs, as required by the challenged claims. However, both the Petition and the supporting expert declaration fail to establish that Welch’s references to “paths” are equivalent to “application programs” within the meaning of the claims. The claims explicitly require assessing the state and health of three

distinct components: hosts, networks, and application programs. Yet, Petitioner and its expert rely on Welch’s disclosure of the health of a path — defined as an end-to-end data and control flow spanning multiple components, including applications, networks, and hardware—and attempt to equate the health of such a complex multi-component flow with that of a software application buried somewhere within the flow.

Welch itself, as well as “Specification and analysis of dynamic, distributed real-time systems,” cited in Welch, clearly distinguishes between application programs and paths. For example, Welch explains that when a path becomes unhealthy, the system may respond by moving or replicating an application program—an action that would make no sense if a path and an application program were equivalent. Because Petitioner offers no meaningful explanation for how Welch’s multi-component “paths” satisfy the claim requirement of evaluating the health of distinct application programs, the Petition fails to demonstrate a reasonable likelihood of success.

a) Petitioner Fails to Establish That Welch’s “Paths” Correspond to the Claimed “Application Programs” Whose State and Health Must Be Determined

The challenged claims require determining the “state and health” of *three distinct* components: (1) the hosts, (2) the *network*, and (3) the *application* programs.

EX1001, 56:60-64.

The Petition states “The resource manager determines whether the paths (i.e., applications) are healthy or not as well.” Petition, 56. However, neither Petitioner nor Petitioner’s Expert offers any explanation for why a “path” discussed in Welch is equivalent to an “application program” in the challenged claims.

Instead, Petitioner’s Expert states that in Welch, “The application programs of real-time control paths send timestamped events to the metrics calculation component, which calculates path-level QoS metrics and sends them to the QoS monitor” and “If a path becomes unhealthy (misses its deadline), the QoS Monitors detect such a condition, diagnose the cause of poor health, and suggest an action (such as moving or replicating an application program) to the Resource Manager.” EX1003, ¶188 (internally quoting Welch, 2-3). From this discussion of “paths,” Petitioner’s expert concludes “the resource manager determines the state and health of the RTCS applications.” EX1003, ¶189.

Yet, Welch makes a clear distinction between “paths” and “program applications.” Specifically, Welch states: “

Our approach is based on the dynamic path paradigm. A path-based real-time subsystem (see [11]) typically consists of a detection & assessment path, an action initiation path and an action guidance path.” EX1005, 1.

Reference [11] corresponds to “L. R. Welch, B. Ravindran, B. Shirazi and C. Bruggeman, “Specification and analysis of dynamic, distributed real-time systems,” in Proceedings of the 19th IEEE Real-Time Systems Symposium, 72-81, IEEE

Computer Society Press, 1998.” EX1005, 5. As discussed above and shown in EX2007, the “Specification and analysis of dynamic, distributed real-time systems” cited is “Specification and **Modeling** of Dynamic, Distributed Real-time Systems” (Ravindran EX1006).¹

Dr. Ranka explains a central concept in “Specification and Modeling of Dynamic, Distributed Real-Time Systems” is the “dynamic real-time path paradigm.” EX2008, ¶108. Specifically, “Specification and analysis of dynamic, distributed real-time systems” states that:

A path-based real-time subsystem (see Figure 1) typically consists of a situation assessment path, an action initiation path and an action guidance path. The paths interact with the environment via evaluating streams of data from sensors, and by causing actuators to respond (in a timely manner) to events detected during evaluation of sensor data streams.

¹ Petitioner also agrees that “Ravindran is document [11] referenced in Welch)” Petition, 39.

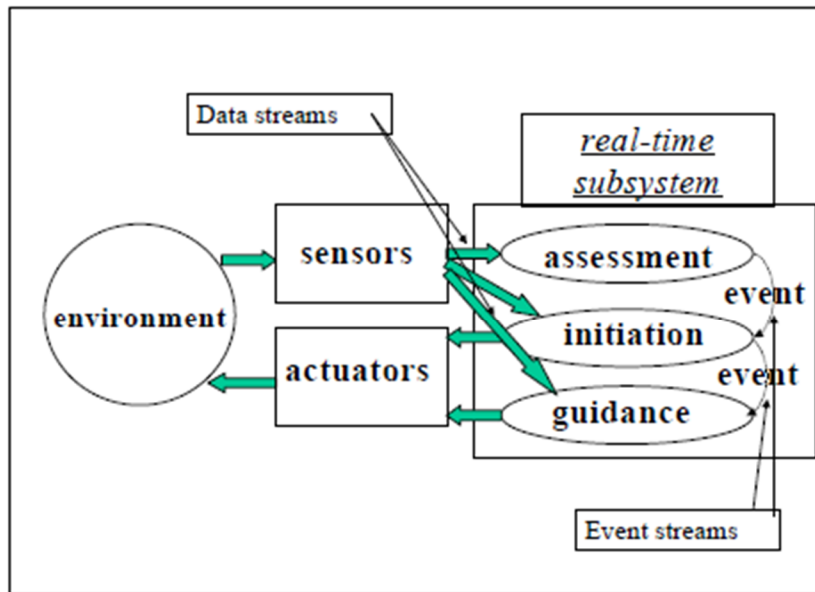


Figure 1. A real-time subsystem.

EX1006, 2

Although Ravindran uses the term “events” it is referring to results of processing the radar data through “assessment,” “initiation,” and “guidance” real time paths to determine whether or not to fire a missile. For example, the assessment path “continuously evaluates the elements of a” sensor data stream to determine if environmental conditions are such that an action should be taken (if so, the action initiation path is notified).” EX1005, 2. Similarly, the initiation path “uses inputs from sensors to determine which actions should be taken and how the actions should be performed, notifies actuators to start performing the actions, and informs the action guidance path that an action has been initiated.” EX1005, 3. Finally, the “guidance path” “is activated by the missile launch event” and “[o]nce activated, the path continuously issues guidance commands to the missile until it detonates (the

deactivation event).” EX1005, 3. Thus, the “events” in Ravindran refer to things like detecting a target, firing a missile at the target and controlling the guidance of the missile to hit the target, not to the monitoring, reporting, or correlation of internal application program statuses or performance metrics as required by the challenged claims. EX2008, ¶110.

Ravindran further clearly distinguishes “application programs” from “paths.” EX2008, ¶110. Specifically, Ravindran states:

Rather than providing real-time support within a particular application language, **it provides support for expressing timing constraints for systems of application programs,** written in a melange of programming languages. **Unlike previous work, in which timing constraints are associated at a relatively small granularity (e.g., task-level), our language allows timing constraints to be expressed the granularity of end-to-end paths which span many communicating programs.**

EX1006, 1-2.

Accordingly, in this context, a “path” is a complete end-to-end flow of data and control through a distributed system, spanning multiple software applications, networks and hardware components (e.g., sensors, processors, actuators). Each path represents a functional pipeline responsible for handling a specific real-time objective, such as situation assessment, action initiation, or action guidance. For example, in a naval missile defense scenario, a situation assessment path would continuously monitor radar data to detect threats, an action initiation path would

coordinate missile launch, and an action guidance path would issue in-flight guidance commands to the missile. EX2008, ¶111.

The usage of the term “path” in “Specification and analysis of dynamic, distributed real-time systems,” is consistent with its usage in Welch. For example, Welch states “[t]he paths interact with the environment via evaluating streams of data from sensors, and by causing actuators to respond (in a timely manner) to events detected during evaluation of sensor data streams.” EX1005, 1. Accordingly, when Welch states “Our approach is based on the dynamic path paradigm,” Welch means they are using the term “path” to describe end-to-end, real-time data and control flows that traverse multiple distributed components—including software applications, networks, sensors, processors, and actuators. EX2008, ¶112.

Neither Petitioner nor its expert offers any explanation for how “end-to-end, real-time data and control flows that traverse multiple distributed components—including software applications, networks, sensors, processors, and actuators,” is an “application program” with the meaning of the ‘743 Patent.

b) Petitioner’s Expert Improperly Equates “Paths” with “Application Programs” Despite Welch’s Clear Distinction Between the Two

This flaw in Petitioner’s mapping is further highlighted by ¶190 of Petitioner’s expert declaration, which states:

190. The resource manager determines whether the paths (i.e., applications) are healthy or not as well. “If a path becomes unhealthy (misses its deadline), the QoS Monitors detect such a condition, diagnose the cause of poor health, and suggest an action (such as moving or replicating an application program) to the Resource Manager (step 8 of Figure 3).” EX1005, Welch, 3.
EX1003, ¶190.

In the first sentence, the expert equates paths and applications as indicated by the “i.e.” However, in the very next sentence, the expert quotes a sentence of Welch that says, “If a path becomes unhealthy.... diagnose the cause of poor health, and suggest an action (such as moving or replicating an application program).” If path and application were equivalent, as alleged by the expert, the sentence in Welch would read, “If a path becomes unhealthy.... diagnose the cause of poor health, and suggest an action (such as moving or replicating the path).” Yet, clearly this is not what Welch says. EX2008, ¶114-115.

Instead, Welch’s statement that “If a path becomes unhealthy (misses its deadline), the QoS Monitors detect such a condition, diagnose the cause of poor health, and suggest an action (such as moving or replicating an application program) to the Resource Manager” further reinforces that Welch is adopting the dynamic path paradigm of “Specification and analysis of dynamic, distributed real-time systems,” and is expressing timing constraints for end-to-end paths which span many communicating programs, network components and actuators. EX2008, ¶114-115.

Petitioner has not shown that Welch discloses or suggests determining the “state and health” of distinct application programs, as required by the challenged claims, which expressly call for evaluating the state and health of three distinct components: hosts, the network, and application programs. Petitioner and its expert improperly attempt to equate Welch’s use of the term “path” with “application program,” despite the fact that Welch makes clear that a “path” refers to an end-to-end control flow spanning multiple components—including software applications, networks, sensors, processors, and actuators. EX2008, ¶68.

This distinction is underscored by Welch’s own language, stating that when a path becomes unhealthy, the system may respond by moving or replicating an application program—confirming that paths and application programs are not equivalent. Petitioner’s expert offers no explanation for how these complex, multi-component paths could reasonably be mapped to individual application programs, and the expert’s inconsistent reasoning only highlights the deficiency in Petitioner’s position. Because Petitioner fails to demonstrate that Welch teaches determining the state and health of application programs as distinct from paths, the Petition does not meet its burden of establishing a reasonable likelihood of success.

3. Petitioner Fails to Show That Welch Discloses QoS Requirements Applied to Individual RTCS Application Programs, Rather Than System-Level Objectives.

Petitioner contends that Welch discloses “QoS requirements dictating parameters regarding service quality of the M management characteristic application programs,” as required by the challenged claims, but neither the Petition nor the expert declaration identifies any disclosure in Welch applying QoS requirements to the RTCS application programs themselves. Instead, the cited portions of Welch describe general system-level objectives—such as dependability, timeliness, and availability—that pertain to the behavior of the overall real-time computing environment, not individual applications. Petitioner’s expert merely recites these high-level system goals. When viewed in context, the cited passages address the operational needs of distributed systems as a whole and are silent regarding application-specific QoS constraints.

Petitioner’s further reliance on Welch’s disclosure of resource reallocation also fails to support the claim limitation, as the resource manager responds to the health of “paths”—complex, end-to-end execution flows—not to individual application performance. Although Welch notes that application programs may be moved or replicated in response to degraded path performance, this reinforces that QoS monitoring is performed at the system level and not targeted to specific applications. Because the cited disclosures describe architectural resource

management and QoS objectives in terms of global system behavior rather than per-application requirements, Petitioner has not shown that Welch teaches or suggests the claimed limitation.

a) Petitioner Fails to Show That Welch Applies QoS Requirements to RTCS Applications Rather Than to the System as a Whole

Petitioner has failed to demonstrate that Welch discloses “QoS requirements dictating parameters regarding service quality of the M management characteristic application programs,” as expressly required by the challenged claims. The cited portions of Welch describe system-level QoS objectives—such as dependability, timeliness, and availability—in the context of overall real-time system behavior, not in relation to specific application programs. Petitioner’s expert declaration at ¶199 merely quotes a high-level discussion of system objectives and offers the conclusory assertion that these objectives apply to the RTCS applications, without citing any portion of Welch that supports such a connection. When viewed in context, the cited Welch passage refers broadly to the design challenges of real-time systems operating in hostile environments and contains no mention of QoS parameters being imposed on, or evaluated in connection with, the Real-Time Control System (RTCS) applications.

The Petition alleges Welch meets “QoS requirements dictating parameters regarding service quality of the M management characteristic application programs,”

because “dependability, timeliness, and availability are examples of a plurality of QoS requirements regarding the quality of the RTCS applications.” Petition, 58. In reaching this conclusion, Petitioner relies on their expert report (EX1003) at ¶199. However, as shown below, ¶199 of the Expert report is devoid of any connection between the QoS requirements in Welch and the RTCS applications.

199. Second, Welch discloses that the “QoS requirements dictat[e] parameters regarding service quality of the M management characteristic application programs.” EX1005, Welch, 2-3. Welch describes using QoS objectives related to timeliness and dependability (“parameters”). Welch explains that “**real-time systems have rigorous, multidimensional Quality-of-Service (QoS) objectives. They must behave in a dependable manner, respond to threats in a timely fashion and provide continuous availability within hostile environments.**” EX1005, Welch, 1. Thus, dependability, timeliness, and availability are examples of a plurality of QoS requirements regarding the quality of the RTCS applications

EX1003, ¶199.

Clearly, the portion of Welch cited by the expert is referencing the global QoS requirements for the system as a whole and is wholly silent with regard to requirements on the RTCS application. EX2008, ¶117. The fact that Welch is discussing QoS of the system and not the RTCS application is apparent when the portion of Welch cited by Petitioner’s expert is viewed in the full context of the paragraph.

Many real-time systems have rigorous, multidimensional Quality-of-Service (QoS) objectives. They must behave in a dependable manner, respond to threats in a timely fashion and

provide continuous availability within hostile environments. Furthermore, resources should be utilized in an efficient manner, and scalability must be provided to address the ever-increasing complexity of scenarios that confront such systems, even though the worst case scenarios of the environment may be unknown (e.g., see [6]). **This paper describes innovative QoS and resource management technology for such systems.**
EX1005, 1

As clearly shown above, the phrase RTCS application does even appear, and the passage describes global, system level QoS objectives. Therefore, there is absolutely no disclosure in Welch that a QoS requirement is applied to the RTCS application. EX2008, ¶117-119.

Petitioner appears to recognize that there is no teaching in Welch that the QoS requirements are applied to the RTCS application. Specifically, the Petition states:

Again, Welch discloses that “[t]he reallocations made by the resource manager make use of information provided by the hardware and software monitors, as well as from a specification file that describes **QoS requirements and the structures of the software system** and the hardware system.” EX1005, Welch, 2. Petition, 59 (also EX1003, 104) (emphasis original)

Again, neither the Petitioner nor the expert has offered any explanation of how the QoS requirement identified in Welch is in any way connected to the RTCS application. As a result, Petitioner has not established that Welch discloses “QoS requirements dictating parameters regarding service quality of the M management characteristic application programs,” as required by the challenged claims. The portions of Welch cited by both the Petition and Petitioner’s expert describe QoS

requirements in the context of the system as a whole—emphasizing objectives like dependability, timeliness, and availability at the global infrastructure and architectural level, not at the level of individual application programs.

Critically, the cited passages are entirely silent with respect to any QoS parameters being imposed on, or tailored to, the RTCS applications. Petitioner's expert offers only a conclusory assertion that such system-level QoS attributes apply to the RTCS applications, without identifying any disclosure in Welch that supports this conclusion. When the cited language is properly read in context, it becomes clear that the discussion pertains to the operational goals of distributed real-time systems as a whole, and does not suggest, let alone teach, the application of QoS requirements to individual software components. Because neither the Petition nor the expert establishes the required link between the QoS requirements and the RTCS application programs, this limitation is not met.

b) Petitioner's Reliance on Welch's Resource Reallocation Disclosures Fails to Show That QoS Requirements Are Applied to Individual Application Programs

Furthermore, Petitioner's reliance on Welch's disclosure that the resource manager reallocates resources based on QoS information does not cure this deficiency. The specification file referenced by Welch describes the structure of the software and hardware systems generally, not QoS parameters specific to the RTCS

application. Similarly, the cited statement that the resource manager may move or replicate an application program in response to the health of a “path” reinforces that QoS monitoring occurs at the level of end-to-end execution flows, not at the level of discrete application programs. See Petition, 58-60.

Specifically, the Petition parrots the expert report and concludes that “the functions of Welch’s software that allocate and reallocate resources to match QoS requirements are thus a “second function” as claimed.” Petition, 59-60 and EX1003, ¶200. In reaching this conclusion, the Petition and the Expert report cite Welch’s disclosure that “If a path becomes unhealthy (misses its deadline), the QoS Monitors detect such a condition, diagnose the cause of poor health, and suggest an action (such as moving or replicating an application program) to the Resource Manager (step 8 of Figure 3).” Id. However, the citation further demonstrates that the health of the path that is what is being monitored, not the health of the RTCS application.

As a result, Petitioner’s reliance on Welch’s disclosure regarding resource reallocation in response to QoS information fails to establish that QoS requirements are applied to individual RTCS application programs, as required by the challenged claims. The specification file referenced in Welch describes the overall structure of the software and hardware systems and does not disclose application-specific QoS parameters. Moreover, the cited passage stating that the resource manager may move or replicate an application program in response to the health of a “path” only

reinforces that Welch monitors QoS at the level of system-wide execution flows—not at the level of discrete application programs.

As explained by Dr. Ranka, Paths, as used in Welch, “paths refer to complex, end-to-end data and control flows that span multiple components, including applications, networks, and hardware, so monitoring these paths’ health is not the same as monitoring the QoS of specific applications.” EX2008, ¶121. As such, the Petition’s assertion that Welch’s resource management functionality constitutes the claimed “second function” is unsupported. Because the cited disclosures address QoS in terms of system-level behavior rather than per-application requirements, Petitioner has not shown that Welch teaches or suggests the claimed limitation.

4. Petitioner Fails to Show That Welch Discloses the Distinct Step of “Configuring” Application Programs, as Required by the Challenged Claims.

The Petition fails to demonstrate that Welch discloses any function that enables “configuring ... the M managed characteristic application computer programs.” The challenged claims recite “starting,” “stopping,” and “configuring” as distinct limitations, each of which must be afforded independent meaning. However, the Petition does not identify any disclosure in Welch describing the act of ‘configuring’ as that term is used in the ‘743 Patent. Instead, Petitioner relies on Welch’s general discussion of starting, stopping, or moving programs and references to system-level configuration displays—none of which equate to “configuring” (or

setting up) an application for execution within a distributed environment. As explained by Dr. Ranka, “configuring” involves a technical setup process that includes specifying things like runtime parameters, and determining execution behavior.” EX2008, ¶132. The Petition not only fails to address these technical steps but also improperly collapses “configuring” into the other claim terms, rendering it redundant. Because the Petition offers no evidence that Welch teaches or suggests this distinct limitation, it fails to meet its burden.

a) Petitioner Fails to Show That Welch Discloses “Configuring” Application Programs as Required by the ‘743 Patent.

Petitioner has failed to demonstrate that Welch discloses “configuring ... the M managed characteristic application computer programs,” as required by the challenged claims. The Petition merely repeats the expert’s reliance on Welch’s discussion of starting, stopping, replicating, or moving programs, but offers no evidence that Welch teaches or suggests the act of “configuring” as that term is used in the ‘743 Patent. None of the cited passages from Welch include any form of the word “configure,” and the isolated reference to the Human Computer Interface (HCI) “provid[ing] information ... regarding the system configuration” refers merely to displaying the current state of the system—not to actively setting parameters in connection with preparing applications for execution. EX2008, ¶124.

Specifically, the Petition again parrots the expert declaration which states:

The program control can “start” and “configure” a program on a particular host. EX1005, Welch, 3. “The program control (PC) component consists of a central control program and a set of startup daemons. When resource manager needs to **start** a program on a particular host, it informs the control program, which notifies the startup daemon on that host.” EX1005, Welch, 3. “The Resource Manager needs to decide on which host(s) and LAN(s) the unhealthy sub-path needs to be **replicated or moved**.” EX1005, Welch, 3. “The HCI provides information to the user regarding the system configuration, application status, and reallocation decisions.” EX1005, Welch, 3. The software can also “shed a program” (“stop”) an application. EX1005, Welch, 2. EX1003, ¶218, Petition, 64-65 (emphasis and marking original)

As show above, the Petitioner and Petitioner’s expert emphasized the words “start”, “replicated or moved” in the excerpts from Welch. Notably absent is any identification of any variant of the word “configure.”

Instead, Petitioner spuriously cites the Human Computer Interface (HCI) of Welch as teaching the configuring of the M managed characteristic application computer programs. However, Welch states “The HCI **provides information to the user regarding the system configuration**, application status, and reallocation decisions.” EX1005, 3. Although the quoted passage does contain the word “configuration” it describes that the HCI displays the information regarding the system’s current configuration.

As explained by Dr. Ranka, “simply displaying the current configuration of a system is not “**configuring** ... the M managed characteristic application computer programs” as recited in the challenged claims. EX2008, ¶132. Instead, in the context

of the ‘743 Patent, the term “configuring” refers to the process of setting up an application to run properly on a host within a distributed computing environment. For example, the configuring may include setting environment variables required by the application at runtime, specifying the working directory, and identifying the correct executable path. Other examples also involve providing necessary command-line arguments, some of which may be dynamically resolved at runtime—such as hostnames of dependent applications, the current date, or user-specific identifiers. Additionally, example of configuring include determining whether the application should run in a terminal emulator (such as an Xterm), as well as defining how the application should be shut down, for example, using a particular script or signal. Beyond these basic setup tasks, examples of configuring also encompasses defining, specifying the resources the application is expected to consume (e.g., CPU, memory, or network), and ensuring compatibility with the host’s operating system and hardware. EX2008, ¶131.

As a result, Petitioner has not met its burden to show that Welch discloses the claimed limitation of “configuring ... the M managed characteristic application computer programs.” The Petition relies solely on expert testimony that repeats language from Welch referring to starting, stopping, or relocating programs, but never identifies any disclosure involving configuration as that term is used in the ‘743 Patent. Critically, none of the cited passages in Welch use the term “configure,”

nor do they describe the specific technical actions associated with configuration—such as setting environmental variables, or specifying command-line arguments. Petitioner’s reliance on Welch’s reference to the Human Computer Interface is similarly unavailing, as the HCI merely displays information about the system’s current configuration; it does not perform configuring of any application computer programs. Accordingly, because the Petition fails to identify any portion of Welch that teaches or suggests the claimed configuration of application computer programs, this limitation is not met, and institution should be denied.

b) The Petition Renders the Term “Configuring” Redundant, Contrary to Federal Circuit Precedent Requiring Distinct Claim Terms to Have Distinct Meanings.

As the Federal Circuit has made clear, “[i]n the absence of any evidence to the contrary, we must presume that the use of these different terms in the claims connotes different meanings.” *CAE Screenplates, Inc. v. Heinrich Fiedler GmbH & Co. KG*, 224 F.3d 1308, 1317 (Fed. Cir. 2000). Thus, “the use of two terms in a claim requires that they connote different meanings.” *Applied Med. Res. Corp. v. US Surgical Corp.*, 448 F.3d 1324, 1333 n.3 (Fed. Cir. 2006). The challenged claims here use three distinct terms: “starting,” “stopping,” and “configuring.” Each term must, therefore, be given a distinct meaning and cannot be treated as redundant.

Despite this clear guidance, Petitioner argues that Welch discloses all three claim limitations merely by restarting an application on a host. Specifically, the Petition alleges that “Welch’s Program Control ... starts, stops, and configures selected ones of the M managed characteristic application computer programs on respective ones of the N hosts in the resource managed system because the Resource Manger finds the host(s) and LAN(s) with the smallest load and then re-starts the terminated program on that host by informing the Program Control (step 5 of Figure 3) and RM HCI (step 6 of Figure 3) accordingly.” Petition, 65 (internal markup removed). However, Petitioner offers no explanation as to how this restart process separately satisfies the requirement for “configuring,” and instead conflates “configuring” with the act of “starting” and/or “stopping.”

Such an interpretation is legally improper. If “configuring” is satisfied merely by starting or stopping an application, the term “configuring” would serve no independent purpose in the claim. But as the Federal Circuit has cautioned, “[i]t is highly disfavored to construe terms in a way that renders them void, meaningless, or superfluous.” *Intel Corp. v. Qualcomm Inc.*, 21 F.4th 801, 810 (Fed. Cir. 2021). Petitioner’s reading does exactly that—collapsing three distinct limitations into one and stripping “configuring” of any independent significance. Because the claims require that “configuring” carry a distinct meaning from “starting” or “stopping,”

and because Petitioner fails to identify any teaching in Welch that satisfies that distinct meaning, this limitation is not met.

Therefore, Petitioner's argument that Welch satisfies the distinct "configuring" limitation merely by restarting an application improperly collapses multiple claim terms into a single action, rendering "configuring" redundant and without independent meaning. By failing to distinguish "configuring" from "starting" or "stopping," and offering no explanation of how Welch teaches the specific technical act of "configuring" as used in the '743 Patent, Petitioner has not met its burden. Accordingly, the Petition fails to establish that this claim limitation is disclosed or suggested in the prior art.

c) Petitioner Fails to Show That Welch Discloses the Distinct Step of "Configuring" Application Programs as Required by the Challenged Claims.

Petitioner has failed to demonstrate that Welch discloses or suggests the limitation of "configuring ... the M managed characteristic application computer programs," as required by the challenged claims. While the Petition repeatedly references Welch's discussion of starting, stopping, or relocating programs, it provides no evidence that Welch teaches the specific technical process of configuration as understood in the context of the '743 Patent. In particular, none of the cited passages in Welch use the term "configure," nor do they describe any actions that constitute configuration—such as setting environment variables,

specifying runtime parameters, or defining command-line arguments. Petitioner’s reliance on Welch’s Human Computer Interface, which merely displays the system’s current configuration, does not fill this gap. As Dr. Ranka explains, simply reporting the configuration of a system is not equivalent to actively configuring application programs within a distributed computing environment. EX2008, ¶132.

Furthermore, the Petition improperly collapses the distinct claim terms “starting,” “stopping,” and “configuring” into a single action—namely, restarting an application. This approach is legally flawed. The claims recite each term separately, and each must be given independent significance. Petitioner’s failure to distinguish “configuring” from the other two terms violates the principle that claim language should not be rendered redundant or superfluous. As established by the Federal Circuit, different claim terms are presumed to have different meanings, and interpreting “configuring” as being satisfied merely by the act of starting or stopping an application would strip that term of its intended function within the claim. Petitioner’s conclusory assertions, unsupported by any evidence in the cited reference, do not overcome this fundamental claim construction requirement.

Because Petitioner has not identified any disclosure in Welch that teaches or suggests the distinct technical function of configuring application programs—and instead attempts to satisfy this limitation by relying on unrelated system behaviors or generic control mechanisms—the Petition fails to meet its burden. The challenged

claims expressly require configuring the M managed characteristic application programs, not merely initiating or terminating them. Accordingly, Petitioner has failed to meet its burden of demonstrating a reasonable likelihood of success.

5. The Suggested Addition of Ravindran does not cure the shortcomings of Welch

The Petition does not allege that Ravindran discloses a second function group providing application event correlation capabilities nor determining the state and health of distinct application programs. Further, the Petitioner does not allege that Ravindran describes applying QoS requirements to individual RTCS application programs rather than to the system as a whole, nor performing the distinct step of configuring application programs.

Instead, the only citation to Ravindran in the Petition is that “Ravindran (Ex.1006) describes hardware monitors as follows: **Hardware (HW) Monitors** (which run on each host) periodically **calculate load indices for hosts and LANS**, and pass this information to the Hardware Data Repository (HDR).” Ex.1006, 7; Ex.1003, ¶147.” Petition, 38 (citations and emphasis original). This disclosure merely pertains to hardware resource monitoring—specifically, the calculation of host and network load metrics for performance tracking at the hardware level. It does not involve any monitoring, evaluation, or management of application programs, nor does it discuss the correlation of application-level events such as statuses, performance metrics, or internal states, as required by the challenged claims.

Likewise, there is no indication that Ravindran discloses the distinct act of configuring application programs as required by the challenged claims. EX20087, ¶134

Accordingly, Ravindran does not remedy the deficiencies in Welch, nor does it supply the missing limitations necessary to establish the unpatentability of the challenged claims.

6. Petitioner Fails to Establish Unpatentability Because Welch and Ravindran Do Not Disclose the Claimed Event Correlation, Application Health Determination, Application-Specific QoS, or Configuring Functions.

Petitioner has failed to carry its burden to demonstrate the unpatentability of the challenged claims. First, Petitioner has not shown that Welch discloses or suggests a second function group providing application event correlation capabilities, as required by the claims. The Petition improperly relies on Welch's discussion of radar-based track correlation algorithms, which pertain to missile tracking in a military system and have no connection to application-level event reporting or correlation. Moreover, Petitioner's alternative theory based on inter-application communication is deficient, as mere communication between applications does not inherently establish the existence of event correlation functionality.

Second, Petitioner has not shown that Welch discloses determining the state and health of distinct application programs. Instead, Petitioner erroneously equates

Welch’s disclosure of monitoring the health of “paths”—complex, end-to-end flows encompassing multiple components—with assessing the health of individual applications. As made clear by Welch and its cited references, paths are fundamentally different from application programs, and Petitioner offers no credible explanation to bridge this critical gap.

Third, Petitioner has not shown that Welch applies QoS requirements to individual RTCS applications, as required by the challenged claims. The cited portions of Welch discuss global system-level objectives such as dependability, timeliness, and availability, not application-specific QoS constraints. Petitioner’s reliance on resource reallocation in Welch further confirms that QoS monitoring occurs at the level of the system as a whole, based on the health of paths, rather than at the level of discrete application programs.

Fourth, Petitioner has failed to demonstrate that Welch discloses the distinct step of configuring application programs, as separately recited in the claims. None of the cited disclosures in Welch describe the technical act of configuration, such as setting environment variables or defining execution parameters. Petitioner improperly collapses “configuring” into “starting” and “stopping,” contrary to Federal Circuit precedent requiring distinct claim terms to be given distinct meanings.

Finally, Petitioner's reliance on Ravindran does not cure the deficiencies in Welch. The Petition only cites Ravindran for the existence of hardware monitors calculating load indices for hosts and LANs, a disclosure entirely unrelated to application event correlation, application-specific QoS enforcement, determining the health of application programs, or configuring applications. Ravindran therefore, fails to supply any of the missing claim limitations.

B. Ground 2 Is Deficient Because Hosokawa Does Not Cure the Shortcomings of Welch and Ravindran

The Petition's reliance on Hosokawa does not remedy the shortcomings of Welch and Ravindran, nor does it establish unpatentability of the challenged claims. As the Petition itself acknowledges, Hosokawa is cited only as "additional evidence as to the obviousness of limitation 10.8" and to provide examples involving numerical relationships between M and N (e.g., $M > N$, $M = N$, or $M < N$). Petition, 79. Hosokawa is not cited for, and does not disclose, any of the missing limitations identified above, including application-level event correlation capabilities, determination of the state and health of distinct application programs, application-specific QoS enforcement, or the distinct act of configuring applications.

Moreover, the Petition's assertion that it would have been obvious for Welch's QoS management software to be stored on one of the "N networked hosts" as taught by Hosokawa is irrelevant to the deficiencies in Welch's disclosures. Whether or not Welch's software is stored on a networked host has no bearing on

whether Welch teaches or suggests the critical functional limitations expressly required by the challenged claims. Hosokawa's disclosure regarding software storage does not bridge the fundamental gaps in Welch's teachings related to application event correlation, application health monitoring, QoS at the application level, or application configuration. EX2008, ¶136-137.

Accordingly, because Hosokawa does not address, let alone cure, any of the substantive shortcomings in Welch and Ravindran, the Petition fails to establish a reasonable likelihood that the challenged claims are unpatentable.

VI. SUMMARY OF ARGUMENTS ON THE MERITS

The Petition fails to demonstrate a reasonable likelihood that the challenged claims of the '743 Patent are unpatentable under either Ground 1 or Ground 2. As explained in detail above, the Petition fundamentally mischaracterizes the disclosures of Welch, and relies extensively on conclusory and unsupported expert testimony that fails to fill critical gaps in the asserted combinations.

First, the Petition does not establish that Welch discloses or suggests a second function group providing application-level event correlation capabilities, as required by the challenged claims. Welch's radar-based track correlation and filtering algorithms, as confirmed by "Challenges in Engineering Distributed Shipboard Control Systems" (EX2006), pertain to tracking and managing missile threats based on radar sensor data, not to collecting and interpreting application program events

such as statuses, performance metrics, or error conditions. Petitioner’s attempt to equate radar tracking with application event correlation is unsupported and contrary to the actual teachings of Welch.

Second, the Petition fails to show that Welch determines the state and health of distinct application programs. Welch describes monitoring the health of “paths” that encompass multiple distributed components—applications, networks, sensors, and actuators—not individual applications. Paths and application programs are distinct concepts, as confirmed both in Welch and in the underlying Ravindran reference (EX1006/EX2007). Petitioner offers no credible explanation for equating these fundamentally different entities.

Third, the Petition fails to establish that Welch applies Quality-of-Service (QoS) requirements to individual managed characteristic application programs. Instead, the QoS requirements in Welch pertain to broad system-level objectives such as dependability, timeliness, and availability, without any indication that these requirements are tailored to or evaluated at the level of individual applications. Petitioner’s expert offers only conclusory assertions unsupported by citation to Welch, which cannot satisfy Petitioner’s burden.

Fourth, the Petition fails to demonstrate that Welch discloses the distinct act of configuring applications as required by the claims. The challenged claims separately recite “starting,” “stopping,” and “configuring” application programs,

each of which must have independent meaning. However, Petitioner conflates configuring with merely starting or moving applications, which improperly renders the “configuring” limitation superfluous. No evidence from Welch supports the necessary acts of configuration, such as setting runtime parameters, or environment variables.

Moreover, Petitioner’s reliance on Ravindran does not cure the deficiencies in Welch. Ravindran’s disclosure is limited to hardware monitoring and does not address application event correlation, application health monitoring, application-specific QoS enforcement, or application configuration.

The addition of Hosokawa in Ground 2 likewise fails to remedy these defects. Hosokawa is cited merely for aspects relating to software storage and the relative values of M and N, not for any of the critical missing limitations. Hosokawa does not disclose application-level event correlation, determination of application health, enforcement of QoS requirements on individual applications, or application configuration.

Finally, the Petition’s extensive reliance on an expert declaration (EX1003) that merely parrots the Petition’s conclusory assertions, with minimal independent analysis or technical reasoning, further underscores the weakness of Petitioner’s position. As recognized by both the USPTO and the Federal Circuit, such repetitive and conclusory expert testimony is entitled to little evidentiary weight.

In short, neither Welch nor Ravindran, individually or in combination, discloses or suggests all of the limitations required by the challenged claims. Hosokawa fails to supply the missing elements. The Petition's arguments are based on a misreading of the prior art, improper speculation, and conclusory expert opinions. Accordingly, the Petition fails to establish that there is a reasonable likelihood that Petitioner would prevail with respect to any challenged claim, and institution should be denied.

VII. CONCLUSION

For the above reasons, Patent Owner respectfully requests that the Board deny institution of the Petition.

Respectfully submitted,

Dated: April 29, 2025

By: /Brandon R. Theiss/
Brandon R. Theiss
Registration No. 70,507
Counsel for Patent Owner
University of Rochester
(Intellectual Ventures II LLC)

CERTIFICATE OF COMPLIANCE WITH WORD COUNT

Pursuant to 37 C.F.R. § 42.24(d), I certify that this **PATENT OWNER'S PRELIMINARY RESPONSE** complies with the type-volume limits set forth in the Memorandum and 37 C.F.R. § 42.24 because it contains 13,154 words, excluding the parts of the Brief that are exempted by 37 C.F.R. § 42.24(a), according to the word processing system used to prepare this Brief.

Respectfully submitted,

Dated: April 29, 2025

By: /Brandon R. Theiss/
Brandon R. Theiss
Registration No. 70,507

*Counsel for Patent Owner
Intellectual Ventures II LLC*

CERTIFICATE OF SERVICE

The undersigned certifies that pursuant to 37 C.F.R. § 42.6(e), a copy of the foregoing **PATENT OWNER'S PRELIMINARY RESPONSE** was served via email (as consented to by counsel) on April 29, 2025 to lead and backup counsel of record for Petitioner as follows:

Jonathan R. Bowser
jon.bowser.ipr@haynesboone.com
HAYNES AND BOONE, LLP
2801 N. Harwood Street, Suite 2300
Dallas, TX 75201

Calmann J. Clements
calmann.clements.ipr@haynesboone.com
HAYNES AND BOONE, LLP
2801 N. Harwood Street, Suite 2300
Dallas, TX 75201

Scott T. Jarratt
scott.jarratt.ipr@haynesboone.com
HAYNES AND BOONE, LLP
2801 N. Harwood Street, Suite 2300
Dallas, TX 75201

Ashraf Fawzy
afawzy@tesla.com
TESLA, INC.
1 Tesla Road
Austin, Texas 78725

Dated: April 29, 2025

By: /Brandon R. Theiss/
Brandon R. Theiss
Registration No. 70,507
Counsel for Patent Owner
Intellectual Ventures II LLC