# An Overview of Regulatory and Trust Issues for the Integrated Clinical Environment

John Hatcliff,
Eugene Vasserman
*Kansas State University*
{hatcliff,eyv}@ksu.edu

Sandy Weininger
*US Food & Drug Administration*
Sandy.Weininger@fda.hhs.gov

Julian Goldman
*CIMIT & Partners Healthcare, Inc.*
jgoldman@mdpnp.org

## Abstract

*The Integrated Clinical Environment (ICE) has been proposed as an open platform for integrating heterogeneous medical devices and coordinating their activities to automate clinical workflows. A key element of the ICE vision is that pieces of an ICE system are verified and receive regulatory clearance in a component-wise fashion (instead of a pair-wise fashion) that will enable the correctness and safety of all possible valid configurations to follow as a corollary of the correctness and safety of its components.*

*In this paper, we sketch elements of a possible oversight approach for ICE systems that we believe will allow regulatory agencies to move beyond the pair-wise clearance paradigm that exists today to one that accommodates component-wise clearance while still achieving agency mandates of assuring safety and effectiveness. We describe several concepts that we believe will be needed in this new regulatory paradigm including third-party certification of compliance to ICE interface standards, tool support for development of ICE components and evidence-based regulatory artifacts, and a security framework that will allow for automatic verification that ICE components are being used as intended in the clinical context, and that they has been cleared for safety and effectiveness.*

## 1 Introduction

Historically, medical devices have been developed as monolithic stand-alone units. Though many devices on the market already include some form of connectivity (serial ports, Ethernet, 802.11 or Bluetooth wireless, etc.), connectivity is usually only leveraged to uni-directionally log data/events from these devices. Each device usually functions as a stand-alone system with its own set of sensors to assess patient physiological properties and possibly actuators to deliver treatment to the patient. In situations where devices are integrated, they are "vertitically" integrated by a single manufacturer, and the configuration of the devices in the system is known a priori.

In contrast to the state of affairs in the medical context, the notion of an integrated "system of systems" is increasingly prevalent in other domains. Small- to mid-scale examples include automotive and avionics systems where many microcontrollers, sensors, and actuators interconnect via a communications infrastructure that allows information from each set of sensors to calculate actions of actuators across the entire systems. In larger examples, military command and control systems are increasingly moving from stand-alone, monolithic systems to integrated platforms.

While many within the clinical and medical device community believe that numerous safety and effectiveness benefits can accrue from creating medical systems of systems, progress is hampered by a lack of interoperability standards and architectures for safely integrating and controlling collections of medical devices to accomplish specific clinical tasks. Not only do the individual devices require scrutiny, but the framework for managing their connectivity and communication with each other also needs thorough verification and validation (V&V). Such challenges increase exponentially when dealing with numerous devices connected to the same communication infrastructure, where not all devices may behave correctly, and some may be outright malicious. Lack of clear guidelines for designing integrated systems is driven by uncertainty regarding how one might construct safety and security arguments for collections of cooperating devices when the full suite of inter-device interactions is not fully known a priori.

The emerging Integrated Clinical Environment (ICE) framework (as defined by the ASTM F2761-09 standard [9]) provides one approach to tackling the challenges above. An ICE-compliant implementation can be viewed as a computing platform (architecture, hardware, and software services) that allows heterogeneous medical devices to be integrated to create medical systems for the care of a single high acuity patient, similar in criticality and functionality to Integrated Modular Avionics [5]. ICE provides services that expose data and control aspects of integrated devices to an ICE Supervisor Application. Individual medical devices providing data and administering treatment are connected to a shared network substrate, managed by the ICE Network Controller. This system can manage multiple devices simultaneously based on a set of extensible clinical applications. It can combine input from multiple devices, synthesize it, display it, and possibly take action, instructing devices to alter their behavior based on feedback from other devices. These "interoperability scenarios" are achieved using different configurations of devices in an application invocation. These application modules can themselves be considered "virtual medical devices."

While there are many technical challenges in building frameworks like ICE, developing a market of ICE compatible interoperating components and deploying ICE systems in a clinical context in a cost-effective manner also requires meeting a number of challenges in the policy domain. Current regulatory paradigms are designed primarily to approve single stand-alone devices or collections of devices that are integrated by a single manufacturer that has complete control over all components. For systems that do include some form of limited integration and reconfigurability of components such as central station monitors, current regulatory policy is that each combination of components requires clearance (e.g., a new 510k application must be submitted). For example, if a new type of medical device is added to the central station monitoring system, then the entire system must be recleared. We refer to this regulatory approach as *pairwise-wise* clearance because the central station infrastructure must receive a clearance for each infrastructure/device pair (combination). In the ICE vision, not only will numerous kinds of devices be integrated, the framework is also intended to be *open* in the sense that, as new technologies and devices emerge, they can also be integrated into a previously deployed interoperable framework. Applying a pair-wise clearance approach to ICE would significantly limit the application and marketability of the framework since each new addition of a device and each new clinical application would require each ICE infrastructure implementation to be recleared.

Experience in consumer electronics with interoperability standards such as USB, WiFi, etc. has shown that it is possible to obtain systems that function correctly by taking a *component-wise* approach. Specifically, manufacturers submit their products to third-party certification organizations that verify that the products conform to interfacing and communication standards. These components are then integrated into larger systems/configurations with high degrees of confidence and without the need to verify each possible combination of components. In the critical systems space, Integrated Modular Avionics [5] and the MILS Security architecture [4] are examples where standards-based architectures and interfaces are being used to encourage the development of a commodity market of safety/security critical components. We believe that lessons learned in these frameworks can help in constructing ICE standards that will allow ICE systems are to be verified and cleared in a *component-wise* as opposed to a pair-wise fashion. That is, correctness and safety of a valid configuration can follow as a corollary of the correctness and safety of its components.

In this paper, we sketch elements of a possible oversight approach for ICE systems that could allow regulatory agencies to move beyond the pair-wise clearance paradigm that exists today to one that accommodates component-wise clearance while still achieving agency mandates of assuring safety and effectiveness. We describe several concepts that we believe will be required for the success of such a scheme including (a) third-party certification of compliance to ICE interface standards, (b) tool support for development of ICE components and evidence-based safety and security artifacts, and (c) an automated trust establishment protocol that will enable clinicians to have full trust in the provenance, quality, and safety of medical devices at connection time as ICE configurations are assembled in the clinical context.

The ideas presented in this paper build on and derive from the significant work of the Medical Device Plug and Play (MD PnP) Interoperability program [10]. MD PnP has lead the effort to develop the ICE ASTM standard, developed influential demos [2, 1] to illustrate the clinical benefits of ICE, and worked with key healthcare providers to develop the Medical Device "Free Interoperability Requirements for the Enterprise" (MD Fire) contracting language requirements to promote the adoption of fully interoperable medical devices and systems in support of patient safety. Several of the broad themes presented in this paper also tie into work being performed by the Prototype Regulatory Submission (PRS) Working Group. The mission of PRS Working Group is to describe and illustrate (with mock regulatory submissions) a "component-wise" regulatory approach for ICE-compliant systems. Thus far, the group has focused on identifying levels of functionality of ICE components and developing a road map of mock regulatory submissions to exercise interoperability points within the ICE architecture. In this paper, we complement that work by sketching the broader ecosphere of an envisioned ICE oversight approach by (a) describing the primary roles and expected activities per role, (b) tool support and novel features, and (c) proposing strategies for establishing trust chains within the ecosphere. We anticipate that some of the concepts presented in this paper will flow into PRS reporting as the PRS work moves forward.

## 2 ICE Architecture and Ecosphere Roles

### 2.1 ICE Architecture

This section briefly reviews the primary entities in an instantiation of the ICE framework as introduced in [3] with a focus on entities that are relevant for our discussion of regulatory and component-wise validation issues.

Figure 1 displays the primary ICE entities, with dashed lines indicating points of interoperability. Yet-to-be-developed standards that build on ASTM F2761-09 will define interfaces and processes for compliance testing/validation that will guarantee interoperability at these points.[1] An *ICE configuration* is a particular instantiation of the ICE architecture of Figure 1 with specific medical devices and specific implementations of infrastructure components such as the Supervisor and Network Controller.

**ICE Supervisor** (infrastructure) — The Supervisor can be thought of as a virtual machine that hosts Supervisor Apps. It provides separation/isolation-kernel-like data partitioning (information cannot inadvertently leak between apps, and apps cannot inadvertently interfere with one an-

[1] One of the goals of the NIBIB Quantum project *Development of a Prototype Healthcare Intranet for Improved Health Outcomes* (Julian Goldman (PI) with the other authors among the team members) will be to build a prototype ICE implementation and develop proposals for interface standards.

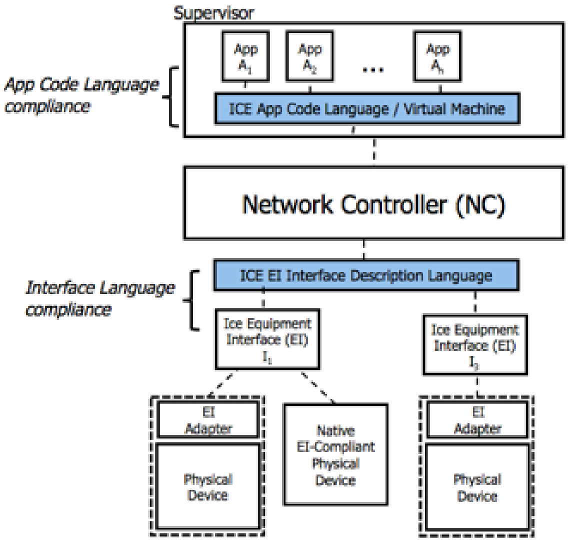
Supervisor
App $A_1$
App $A_2$
...
App $A_n$
App Code Language compliance
ICE App Code Language / Virtual Machine
Network Controller (NC)
Interface Language compliance
ICE EI Interface Description Language
Ice Equipment Interface (EI) $I_1$
Ice Equipment Interface (EI) $I_3$
EI Adapter
Physical Device
Native EI-Compliant Physical Device
EI Adapter
Physical Device

**Figure 1. Primary ICE components and interoperability points with description languages to accommodate interface variability**

other) and time partitioning (real-time scheduling guarantees that the computations in one app cannot cause the performance of another to degrade or fail). The Supervisor also provides a console that allows a clinician to launch apps, monitor their progress, and provide user input during app execution.

**ICE Supervisor Apps** – programs that accomplish a clinical objective by interacting with one or more devices attached to the Network Controller. The nature of apps may range from lightweight scripts that coordinate the actions of devices, to more substantial algorithms that extract physiological parameters from sensor waveforms, or complex control algorithms that implement closed-loop physiological control. In regulatory terms, each app as it executes in the Supervisor defines the "intended use" of the current ICE configuration. An important safety-related concept is that ICE medical devices never interact directly with each other; all interaction is coordinated and controlled via Supervisor Apps.

**ICE Network Controller** (infrastructure) – In essence, the Network Controller is high-assurance middleware that establishes virtual "information pipes" between devices and apps running in the Supervisor. It exposes the ICE Interfaces of attached devices to Supervisor apps, and is agnostic as to the intended use of the clinical apps that it supports. The fact that the Network Controller only forwards data/commands to/from apps and devices and does not transform the data in any way is a key property that enables an argument that it can indeed by cleared "once and for all", i.e, component-wise. The Network Controller also manages the discovery and connection protocol for devices that wish to connect to the system.[2] In Section 4, we will see that an important step in the connection protocol is an *authentication* step which validates the identity of device along with a digital certificate that provides guarantees about its provenance and ICE compliance. Once a device is connected, the Network Controller actively monitors the health of the device connection, and notifies an app when a device that it utilizes has a failed connection.

**ICE Data Logger (not shown in the diagram)** (infrastructure) — A component dedicated to logging of communication and other important events within the Network Controller and Supervisor.

**ICE Equipment Interface (EI)** — The ICE EI declares the functional capabilities of the device, e.g., format of its data streams, commands to which it responds, along with non-functional properties of the data such as the rate at which data elements are streamed from the device, the accuracy of the data, etc. In regulatory terms, the ICE EI captures the intended use of an ICE Compliant Device within ICE.

**ICE Equipment Interface Definition Language (ICE IDL)** – Each ICE EI is specified using the ICE IDL — a formal data description language possessing (a) collection of data types sufficient for describing the format of common clinical data, (b) notations for specifying the form of commands to which a device responds and, more broadly, the types of device interactions that a Supervisor App may take advantage of (e.g., polling of certain data fields, pushing parameter settings to the device), and (c) other semantic constraints or rules that cannot be directly captured by types or simple attributes (this might include a simple state machine describing the allowable order of command calls on the interface). We anticipate that this language would have a notion of sub-typing / sub-classing that would allow a generic interface to be written for a particular category of devices (e.g., pulse oximeters) and the sub-classing would be used to describe optional capabilities (e.g., features that would be available for a particular model of pulse oximeter).

**ICE Compliant Equipment** — A device suitable for integration into an ICE system is either a *Natively Compliant Device* (e.g., the data transformations necessary to convert native device data to the format expected by the ICE Equipment Interface are implemented in the device's hardware/firmware) or an *Augmented Compliant Device* in which transformations necessary to satisfy the ICE Equipment Interface are implemented by an *ICE Equipment Adapter* either as supplemental hardware (e.g., a dongle) on the device side, or as a service on the Network Controller. In the case of an Augmented Compliant Device, for the purposes of regulatory approval, the ICE Equipment to be considered for clearance shall consist of the medical device bundled (paired) with its adapter.

[2] It is possible to imagine the Network Controller providing generic data mediation services that, e.g., adapt the rate at which data is transmitted from devices to apps.

# 3 Regulation Within the ICE Ecosystem

## 3.1 ICE Regulatory Ecosphere Roles

As a foundation for sketching the regulatory structure and associated trust chains, we summarize below the primary roles that we anticipate being involved in the ICE regulatory process.

**Medical Device Manufacturer (MDM)** — the entity designing and/or physically constructing a medical device to be used in a clinical facility. We use this term to refer to manufacturers of traditional medical devices as opposed to companies that manufacture ICE infrastructure components or adapters to incorporate conventional devices into ICE.

**Regulatory Authority (RA)** — the regulatory authority that issues clearances for medical devices and medical systems with the goal of assuring that those devices/systems are safe and effective for their intended use. In addition to approving conventional medical devices for market, the RA has the following mechanisms that can be brought to bear to assure the safety and effectiveness of ICE components:

- Leverage interface standards and process standards produced by the ICE Alliance (defined below)
- Processing applications for "ICE clearance", i.e., submissions that claim that ICE Equipment, ICE Network Controllers, ICE Supervisors, and ICE Supervisor apps are conformant to ICE standards and are safe and effective for use within ICE.

**ICE Equipment Originator** — Note that a MDM may manufacture a medical device but may play no role in making it ICE compliant. In addition, companies may market ICE infrastructure components but not stand-alone medical devices. Thus, we introduce the term ICE Equipment Originator to refer to a person or entity proposing equipment to be integrated into ICE. This would include medical device manufacturers (as defined above) if the manufacturer of a device is the entity responsible for making it ICE compliant, medical device integrators (e.g., a company that specializes in building interfaces for devices), or companies that manufacture ICE components such as Network Controllers, Supervisors, or Apps.

**ICE Alliance** — organization of ICE Equipment Originators and other interested parties who cooperatively develop standards for ICE components sufficient to achieve safe interoperability between component instances. Moreover, since ICE is an open framework whose full range of components is not known a priori, the ICE Alliance will also define processes by which new component instances can be judged to be ICE compliant and thus suitable for integration into an ICE. Specifically, the members of the Alliance will carry out the following tasks cooperatively:

- Develop interface/architecture standards for ICE components such as the Network Controller, Supervisor, Data Logger.
- Develop and maintain a standard for ICE Equipment Interface Definition Language (IDL)
- Develop a standard for clinical app definition and scripting language that is used to specify the behavior of apps hosted by the Supervisor
- Develop a process by which (a) an ICE Equipment Originator desiring to integrate equipment (e.g. a medical device) with ICE may make a submission requesting recognition of compliance to an ICE Equipment Interface, (b) the submission may be judged to be ICE compliant, (c) a certificate of ICE compliance can be issued to the originator to be used as part of the authentication process when the equipment is connected to an ICE Network Controller.
- Develop a process by which an ICE Supervisor app author may make a submission requesting certification of ICE compliance for the app, (b) the submission may be judged to be ICE compliant, (c) a certificate of ICE compliance can be issued to the originator to be used as part of the authentication process when the app is installed in an ICE Supervisor.
- Develop templates and guidelines for consistently structured artifacts for clearance submissions to the Regulatory Authority.
- Develop test suites and evidence-based tools for assessing ICE compliance of proposed ICE components. Note that ASTM F2761-09 requires that the ICE Equipment Originator of a device must include a qualification test suitable for use by the Responsible Organization (defined below) to verify those portions of the basic safety and essential performance of that device that can be affected by the device's ICE Equipment Interface. We do not anticipate that the "qualification test" function would be rich enough to expose all issues necessary to fully evaluate ICE compliance (these anticipated limitations motivate the need for third-party testing and the development of testing infrastructure by the ICE Alliance). ICE testing infrastructure may be used by the ICE Equipment Originator to develop the qualification test functionality.
- Develop and maintain a well-organized library of ICE Equipment Interfaces that (a) ICE Equipment Originators will be expected to target when developing equipment that they intend to be natively ICE compliant or adapters that make legacy device ICE compliant, and (b) developers will target when coding Supervisor Apps.

**Responsible Organization** — a Healthcare Delivery Organization (HDO) (e.g., hospital) responsible for managing an ICE installation. The ICE installation would typically be located at the organization. However, this definition is broad enough to admit situations where an ICE is installed in a home but managed by an overseeing HDO.

## 3.2 Anticipated Market Forces Impacting Regulation

Recent experiences with smart phones as computing platforms (e.g., iPhone and Android) illustrates how well-designed open platforms can encourage innovation and give rise to an explosion in lightweight apps providing highly targeted functionality. Based on this experience, once ICE infrastructure is on the market, there will likely be a flood of ICE apps and device interfaces submitted for marketing clearance — significantly more than the number of Class II & III device approvals that are submitted now. Moreover, the safety issues in open systems are much more challenging. For example, apps will need to work with devices with which they have previously not been tested, there is potential for interference between apps/devices is much greater, it will be easier for "fly by night" operators to "roll their own" apps/interfaces, etc.

Therefore, the existing regulatory process needs to be assessed with the goal of developing strategies for ICE-related submissions to better support (a) increased speed in processing submissions, and (b) increased scrutiny of safety and functional/security claims. While we anticipate that multiple Network Controller and Supervisor instances will be developed and marketed, it is the app and device interface elements of ICE where we anticipate the highest volume by far. Therefore, special attention will be given to guiding development and compliance checking for these elements.

Here are some of the key features of the envisioned ICE compliance and regulatory oversight process.

**Standardized Development and Compliance Validation Processes** — The ICE standard will not only define standards for architecture and interfaces; because ICE is an open platform and ecosphere whose integrity needs to be maintained, the ICE standard must also include well-defined processes recognized by the Regulatory Authority for incorporating new elements into ICE (apps, device interfaces primarily).

**Criticality Levels for ICE Elements** — As with conventional medical devices, apps and interfaces vary in complexity and the level of risk they present. Therefore, criticality classifications will be developed for apps and interfaces (and perhaps other ICE elements) so as to provide a foundation for a clearance process where the degree of regulatory oversight can be commensurate with the level of complexity. For example, an app that simply forwards data to a display embodies less risk than an app that implements a safety interlock, which in turn embodies less risk that an app that implements a closed loop control algorithm. For interfaces, an interface for a monitoring device that exposes no control features for the device embodies less risk than an interface for a ventilator that exposes both data and control features. The design of criticality levels will be inspired in part by connections to existing regulated data systems. For example, a recent FDA ruling [12] reclassified Medical Device Data Systems (MDDS) (systems that transfer or exchange of medical device data from a medical device, without altering the function or parameters of any connected devices) from Class III to Class I devices – so that they no longer require Premarket Approval or Premarket Notification (510k). Thus, one can imagine that boundary for the lowest classification level for interfaces and apps might be designed to align with the definition of an MDDS. The implication is that if a device interface and associated app simply forwards data for display without altering the function or parameters of the associated device), it would not require a clearance.

**Standard Development Environments for ICE Apps and Interfaces** — Similar to the App Development Kits supplied by Apple and Google and Device Driver Development Kit supplied by Microsoft, the ICE Alliance will supply development environments for both Supervisor Apps and IE interfaces. These development environments will include test suites, static analyses, and verification tools that both developers and third-party certifiers apply to validate compliance to ICE interface/app standards — ensuring that elements submitted for clearance have been subjected to a uniform and rigorous validation. Another potential model here is Continua, which provides a variety of code and testing infrastructure to its members.

**Development Tools Codify Standard Development Processes and Regular Structure** — ICE development environments will include functions that guide interface and app developers in constructing regulatory submissions that have a regular structure that can be more easily audited and examined with respect to their safety/compliance claims. The Common Criteria Authoring Environment [6], which provides a suit of template structures and certification document construction aids in the domain of security certification, is a potential model. Because of the direct integration with the development process, we anticipate that these tools will provide support for common activities such as hazard analysis and assurance case construction. For example, upon releasing interfaces for a particular device class, the ICE Alliance development environment would also be seeded with template structures for hazard analysis that include the common hazards for that device class.

**Facilitating the Development of Third-party Certification Agents** — Many app and device interface developers will not have the equipment/resources to test their products thoroughly with multiple Supervisors, Network Controllers, etc. Therefore, we expect that the ICE Compliance Assessment process will include the notion of third-party certification agents (and an associated process for credentialing the agents) who will (a) assess ICE components that manufactures wish to bring to market and (b) judge compliance of proposed components against the ICE Standard. The compliance evaluation process will employ the ICE testing and validation environment described above plus additional ICE installations and tests. Again, having common validation environments will help ensure a more uniform validation process across multiple third-party certifiers. The third party certification concept is important for (a) maintaining the integrity of the ICE ecosphere and (b) reducing the workload of the FDA in the context of a high volume of app and interface submissions. Similar arrangements are used in other domains. For example, wireless devices are submitted for third-party certification to the WiFi Alliance in addition to being submitted to the Federal Communications Commision.

### 3.3 From Pair-Wise to Component-wise Regulatory Clearance

In the envisioned compositional paradigm of ICE, whenever a new configuration of ICE components is introduced in support of an ICE clinical application, that particular configuration does not have to be cleared — clearance is only required for ICE components that have not previously been cleared (note that the notion of "component" used here is intended to include "app code" that controls the interactions between devices and establishes the behavior of ICE clinical application). For example, in a previously cleared application that requires a pulse oximeter, the specific pulse oximeter device from MDM $A$ used in a previous configuration for that application may be replaced with a pulse oximeter from MDM $B$ without re-clearing the Network Controller, Supervisor App or other components in the configuration — a new clearance is only needed for the new device from MDM $B$ to ensure that it is ICE compliant. Similarly, given a specific selection of ICE compliant devices, introducing a new ICE Supervisor app code to provide the realization of a new clinical application does not require reclearing all the utilized devices and ICE infrastructure components — clearance is only needed for the new app code.

Establishing this compositional paradigm requires careful engineering, clear interoperability standards, and a clearly defined process recognized by the regulatory authority by which stakeholders incorporate new components into ICE.

The conventional approach for achieving interoperability in an architecture between two modules $A$ (e.g., Network Controller) and $B$ (e.g., Data Logger) is to define an interface $I_{AB}$ between $A$ and $B$ that precisely defines the possible interactions at that interoperability point. As manufacturers produce different instances $A_1, A_2, \ldots, A_n$ of $A$ and $B_1, B_2, \ldots, B_m$ of $B$, it is not necessary to test/validate all possible combinations $A_iB_j$. Instead, each instance is shown to be compliant to its side of the interface, e.g., $A_i$ complies with $I_{AB}$ and $B_j$ complies with $I_{AB}$. Demonstrating compliance is achieved typically by (a) providing with $I_{AB}$ a pair of test suites $S_A$ and $S_B$ that sufficiently approximates the behaviors of all instances of $A$ and $B$ respectively, (b) testing each instance $A_i$ against the tests in $S_B$ and each $B_j$ against the tests in $S_A$. It is expected that $S_A$ includes a sample of instances $A_i$ that provides "appropriate" coverage of the behaviors of all potential $A$ instances, as well as additional tests that stress potentially problematic dimensions of $B$ behaviors. Ideally, stronger static analysis and formal methods are developed that can give greater confidence and even mathematical guarantees that, e.g., $A_i$ can handle any interactions that could ever occur with any $B_j$.

The discussion above already reveals some important concepts related to a potential regulatory paradigm. First, the interface of a module defines the intended use of its instances within the architecture. Therefore, the interoperability interfaces discussed above should be subjected to elevated regulatory oversight. Second, a potential "hazard" in the process for judging interface compliance sketched above is that the validation process (e.g., included test suites $S_A$, $S_B$ and associated analysis methods) is not sufficiently rigorous for ensuing that, e.g., an instance $A_i$ can perform correctly with any instance $B_j$ (e.g., due to $S_B$ giving rise to inadequate test coverage when exercising $A_i$'s behavior.) Therefore, the compliance validation process itself should be subject to regulatory oversight.

It is useful to recognize that there are two types of interfacing that will need to be used to achieve interoperability at these points: fixed interfacing and variable interfacing. Fixed interfacing occurs when the specific interactions between modules A and B are known a priori — when the ICE standard is completed and before the first ICE with regulatory approval is deployed. Such interfacing can be captured with a single interface $I_{AB}$ and with conformance validation proceeding as outlined above. For example, the interface between the ICE Network Controller and ICE Data Logger will be defined in a subsequent ICE standard and then remain unchanged (with the exception of possible updates to the standard). Thus, once a data logger receives regulatory clearance by demonstrating its compliance to the Data-Logger-Network-Controller interface, there is no need to re-clear it when it is paired with different network controllers. In contrast, interoperability between the Network Controller and devices cannot be achieved through a similar fixed interfacing strategy because (a) different manufacturer models of devices present different interfaces to the network controller and (b) the range of device models that may seek to interoperate with ICE is not known a priori (there will continue to be emerging device types that will be integrated into ICE). A similar situation exists at the interoperability point between the ICE Supervisor and app codes — Each app code will require different data streams and control capabilities for devices. In both of these interoperability points, we desire a solution that allows for variability in the interfacing.

The engineering approach taken by ICE to tackle the problem of variable interfacing centers around the mandatory use of description languages. For example, ICE includes an ICE Equipment Interface Definition Language (described earlier) that must be used to specify the interface of any device that wishes to interoperate with ICE. With this in place, clearance of an ICE Network Controller changes from the problem of validating each Network Controller against all possible devices (the pair-wise problem) to the problem of validating each Network Controller against the ICE Equipment IDL. By adding a (meta)-level of description (the IDL) above the device interface, the IE IDL becomes a fixed interface for the Network Controller (the IE IDL can be completely specified a priori) and each Network Controller only needs to be validated once as correctly supporting the IE IDL. Though technically this still implies that each Network Controller must be shown to correctly handle any device interface written in the IDL, the constrained nature of the IDL and its inductive structure will allow a systematic enumeration of test cases or even formal inductive reasoning to be applied to achieve this validation goal.

The concept is similar to that of an interpreter/compiler: the validation goal for a compiler for language L is to show that it can correctly compile any well-formed L program (and reject programs that are not compliant with L's definition). Since the concept of validated compilers/interpreters is used in other high-assurance domains (e.g., avionics), there is little reason to doubt that it could be applied in this context. The ICE Network Controller can be thought of as an interpreter for the IE IDL. The IDL description for a device is supplied to the Network Controller in a "plug-n-play" step when a device connects and is authenticated (alternatively, the interface can be preloaded by an administrator). The Network Controller "interprets" the IDL description to set up appropriate communication channels for the device to communicate to Supervisor apps. Alternatively, the Network Controller rejects the IDL description because it is not well-formed or because the computational/real-time requirements of the device as described by the IDL exceed the current capabilities of the Network Controller.

This same description language strategy is used to tackle the app code variability in the Supervior / Application Code interoperability point. Specifically, ICE includes an app code language (also referred to as a "scripting language") for specifying applications. With this in place, the Supervisor does not need to be cleared "pair-wise" for each app. Instead, it is cleared once with respect to the app code language. Clearing a Supervisor involves showing that it can properly interpret the functional and non-functional (e.g., real-time) requirements of any app written in the app code language. It is important to understand that judging compliance for both of these types of "interpreters" will be aided by the application of open test suites and other development tools produced by the ICE Alliance.

### 3.4 Proposed Certification / Clearance for ICE Components

We believe a key will be the parallelism in the roles played by the Regulatory Authority and ICE Alliance. Since the ICE Alliance defines the ICE architecture, interfaces, and compliance processes, it can and should be more prescriptive than the Regulatory Authority would usually be about the format of submissions and evidence justifying safety and effectiveness. In addition, given the need to assure interoperability, it is also reasonable for the Regulatory Authority to accept a strong role for third-party certification as is the practice of ensuring interoperability in other domains.

Below we briefly summarize the regulatory submission and ICE compliance goals for each ICE entity.

**ICE Network Controller** — requires third-party certification of ICE compliance and regulatory clearance. The network controller is cleared against a specific version of the ICE interface description language and NOT against a specific list of ICE compliant devices. In the interoperability verification of an Network Controller, the primary task is to verify that it can correctly interpret and support any ICE compliant interface definition by testing or evaluating the network controller against a suite of representative test samples that cover the interface description language features.

**ICE Equipment Interface Definition** — We anticipate that the interface description intended for a single or family of devices would not require regulatory clearance, since it is not a medical device itself. However, the interface and certainly any interface adapter used in an augmented device would be evaluated when a device is cleared for use in ICE. We note however that the ICE Alliance is expected to play an important role in proposing and examining interfaces as stand-alone entities. As described earlier, since it is in the interest of the ICE Alliance to maintain a well-organized library of interfaces following particular style guidelines, the Alliance may propose standard interfaces for different device classes and advise against interfaces that are sufficiently close to an existing interface with only minor differences in functionality.

**ICE Equipment Interface / Equipment Pairing (Binding)** — Each manufacturer model of equipment to be used with ICE must be paired with an ICE Interface written in ICE IDL and (a) submitted to the ICE Alliance for ICE Compliance Certification, and (b) submitted for regulatory clearance to be used as an ICE device (an exception may occur if the interface is simple enough to allow it to be classified as an MDDS device). Note that even though a device has previously received a clearance as a stand-alone device, it may also need a separate clearance for use in ICE. In this case, the device's ICE interface defines the intended use of the device within ICE. A device is cleared for use in ICE against its paired ICE device interface specification and NOT against a specific list of network controllers. The primary interoperability verification task here is to show that the device correctly implements the ICE Equipment Interface with which it is paired.

**ICE Supervisor** — Each ICE Supervisor implementation must be submitted to the ICE Alliance for ICE Compliance Certification, and (b) submitted for regulatory clearance to be used within ICE. The primary interoperability verification task here is to show that the supervisor safely and correctly interprets apps written in ICE App language. There are numerous other safety and security issues related to separation kernel functionality and fail-safe behavior.

**ICE Supervisor Application** — It is anticipated that each app would (a) be submitted to the ICE Alliance for validation of ICE compliance, and (b) some sort of filing with the regulatory authority. The filing may be based on a concept such as the US Food and Drug Administration's 510k, which requires that the submitter to identify a "predicate device" that has similar functionality and safety issues. While 510k clearance will be sufficient for many apps, some apps may require the more stringent Pre-Market Approval (PMA), e.g., apps that implement new forms of closed-loop control.

# 4 Automated Trust Reasoning Within The ICE Ecosystem

Without security, the connection of a single malicious device to an ICE installation can disrupt all aspects of the system: the device may falsify or suppress data from other devices, send rogue commands to connected devices, or prevent all communication entirely, disabling the entire system. Since ICE is an open standard, compatible devices by definition can be built by almost anyone. Therefore ICE may potentially integrate components whose provenance, quality, and safety cannot be immediately ascertained by clinical or technical staff at a clinical facility. It is therefore critical to design security mechanisms such that the point of care infrastructure can leverage trust between the various entities within the ICE ecosphere to validate components in real time, as they are integrated by clinicians.

While there are multiple possible approaches for achieving this goal, in this section we outline one particular approach to establishing initial trust. Both initial device verification and runtime monitoring and enforcement are important, for the sake of brevity we focus on authentication actions that take place at initial device connection.[3] The key is to provide a simple and intuitive way for the clinical facility and ICE to trust the device which is being connected to the network, since the device has the potential to (a) be malicious, (b) incorrectly identify itself as a device from another manufacturer, (c) claim functionality for which the device has not been approved by a regulator or certifier, (d) claim ICE compliance when it fact it has not been through compliance testing by the ICE Alliance, etc.

Considering the unique technological and regulatory challenges (and their interaction), existing security solutions apply, but in a limited way. Earlier work on secure and private medical device interoperability [13] provides a foundation to build on. It is important to consider how these previous solutions can be deployed in the context of a regulatory paradigm. A solution that is not synergistic with the regulatory process is unlikely to be adapted by the regulatory agency, and thus the manufacturers. Bulletproof security, even if achievable, is not always desired; in fact, it is explicitly not allowed under certain circumstances. Consider for instance an infusion pump which only accepts commands that are entered by an authorized doctor who logs in before reprogramming the device. However, if a clinician has decided that immediate and drastic action is needed, that clinician *even if not normally authorized*, must be able to override safety and privacy interlocks otherwise enforced by the coordination system and individual devices. Therefore, we must be very careful in designating the security design "concept of operations" to allow maximal trust, authentication, and privacy, while allowing them to be overridden when required.

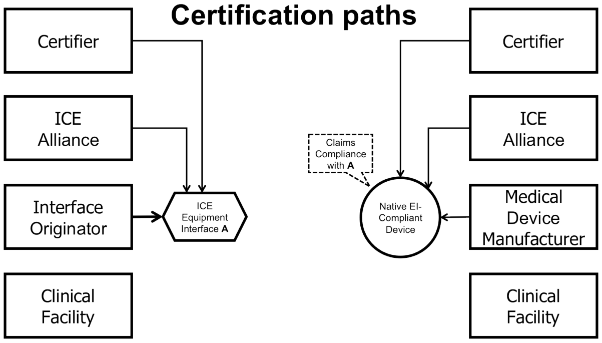

**Figure 2. The certification process, including submission, for (left) an ICE Equipment Interface (EI) and (right) an ICE-compliant device that its manufacturer claims conforms to a specific EI.**

The trust chain concepts that we outline below can be applied to all of the interoperable ICE components including Supervisor and Network Controller, but clinicians will most often be integrating new ICE Supervisory Apps and ICE Equipment. Therefore, we introduce security concepts using ICE Equipment as an example.

Figure 2 shows a potential certification process for devices or ICE Alliance library interfaces. To ensure trust at the point of use (clinical facility), we would need human-readable as well as machine-verifiable credentials. The goal of machine-readable credentials as well as device and interface specifications (self-describing device models) is to enable, with minimal to no human intervention, for the ICE infrastructure to validate and verify (and thus trust) the provenance, integrity, safety, and effectiveness of medical devices (e.g. verified and approved for ICE) when they are connected to the clinical facility's network. Ideally this would involve the Regulatory Authority as the ultimate root of trust (all trust in other parties in the system would derive from it). For this we can use well-known security tools and standards, such as a public key infrastructure (PKI) and certification (e.g. X.509 [8]). However, we can also employ multiple roots of trust in the form of third-party certifiers, analogous to the way SSL/TLS certificates currently work to authenticate websites [7]. For illustration purposes, we will use the multiple certifier model, and assume the notion digital certificates and signatures that have the following properties:

- Each ICE entity, interface, and device instance have unique certificates;
- Each ICE entity and device instance have unique signing keys;
- Signatures cannot be forged;
- Certificates cannot be undetectably altered;
- The signer of a certificate attests to the validity and integrity of information in the certificate;
- The signer of a certificate can be identified from the certificate itself; and
- Signatures are non-repudiable (i.e. having signed a

[3] This likely only happens once, since after a device has been authenticated, information about previous connections and a device's unique identifier (UID) allows an accelerated authentication process for subsequent connections.

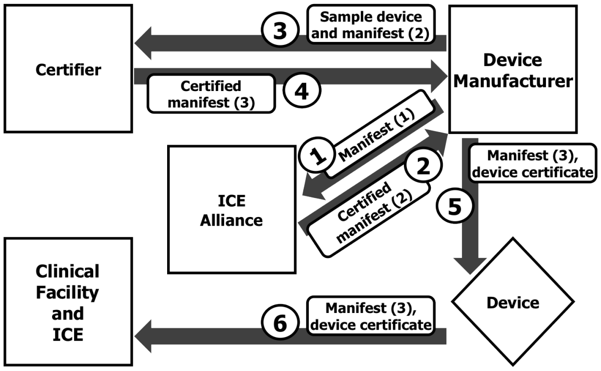
Certifier
3
Sample device and manifest (2)
Device Manufacturer
Certified manifest (3)
4
1
Manifest (1)
2
Certified manifest (2)
Manifest (3), device certificate
5
ICE Alliance
Clinical Facility and ICE
Device
6
Manifest (3), device certificate

**Figure 3. The primary roles in the ICE ecosphere relevant to integration of ICE entities and equipment, and the multiple certificates used to establish a chain of trust from the point of care all the way back to the component certifier.**

certificate, an entity cannot later claim that it did not).

This system allows entities to assert information or trust relationships in a machine-readable format, requiring minimal to no human intervention during verification.

Figure 3 outlines the steps to produce a machine-readable credential for a device that conforms to an existing ICE Alliance-approved equipment interface (EI). For simplicity, we will confine our discussion to "natively-compliant devices" — those suitable for integration into an ICE system without supplemental hardware. This also combines the Medical Device Manufacturer (MDM) and ICE Equipment Originator into a single entity; we will refer to it as "manufacturer." When designed, a device either will either conform to an existing approved EI, or will be submitted for certification together with a new interface description, written in a domain-specific ICE Interface Definition Language (IDL). Since the IDL is standardized, any interface written in IDL will be compatible with ICE, and therefore it is not necessary that each interface be validated and certified by the ICE Alliance. However, it would likely be helpful in the overall certification process, reducing the burden on the certifier, similar to demonstrating ISO process compliance.

In step one, the device is submitted to the ICE Alliance along with a device type manifest (DTM), which is a self-describing machine-readable document that defines the type of device, its technical specifications such as performance guarantees and operating parameters, and the ICE interface that it uses. The DTM also includes the manufacturer's name (or other identity), a copy of the manufacturer's public key, and the manufacturer's signature. The ICE Alliance confirms, via third-party certification, that the device indeed conforms to the claimed interface, signs the DTM with the ICE Alliance private key, and gives the newly-signed manifest to the manufacturer in step two. Since the ICE Alliance public key is known to everyone, anyone can verify that the Alliance signed the manifest. However, since the private key is secret from anyone outside the Alliance, no one else can produce a valid Alliance-signed manifest. In step three, the DTM submits the ICE Alliance-signed manifest to the certifier, along with a physical device. The certifier can confirm that the device has been examined by the ICE Alliance and certified to conform to the claimed interface by verifying the ICE Alliance signature on the DTM. When a device is cleared for use in a clinical facility (within operating and performance criteria specified in the manifest), the certifier signs the device manifest with its private key, giving the resulting certificate to the manufacturer in step four. Step five shows that this final machine-readable device type certificate (DTC) is included with each physical device as part of a device instance certificate (DC), which also includes the unique device instance identity (UID) and the device public key. The DC is signed by the device manufacturer.

When the device connects to the clinical facility's network in step six, it first presents its DC to the facility ICE, which is hard-coded with at least the public keys of all certifiers and the ICE Alliance. By examining the device instance certificate, ICE verifies that:

- The device manufacturer has been approved by a trusted certifier (by checking the certifier's signature on the DTC)
- The device manufacturer's public key in the DTC matches the key that signed the DC (by checking the certifier's signature on the DTC)
- The device is a genuine device from the manufacturer (by checking the manufacturer's signature on the DC)
- The device is operating within its intended environment (by reading operating parameters from the DC)
- The device is performing as expected (by reading performance guarantees from the DC)
- The device conforms to an ICE-compliant interface (by checking the ICE Alliance and/or certifier signatures on the DTC)

This process traces the device, its operating parameters, performance guarantees, and interface conformance from the manufacturer, to the ICE Alliance, to the certifier. Since the certifier has attested to all the details regarding the medical device, the device can be considered safe for use. Note that by signing a manifest certified by the ICE Alliance, the certifier implicitly recognizes the ICE Alliance as an authorized entity to verify device-interface conformance.

If the manufacturer submits a new Equipment Interface as part of device certification, the process above is slightly different. We have previously assumed the EI used by the device is known to ICE. If ICE must "learn" a new interface in order to use the device, that interface itself should undergo the same certification process as the device, and the interface descriptor, signed by the ICE Alliance and certifier, is included with the device. As with the device, ICE can trace certification from the ICE Interface Originator to the ICE Alliance, to the certifier.

## 5 Conclusion

While there are numerous items still to be addressed, we have tried to address the most important points in designing trust mechanisms for ICE systems to be synergistic with possible regulatory approaches and organizations of stakeholders within the ICE ecosphere. On the technical side, important next steps include developing proposals for the ICE Equipment IDL and associated verification processes. Prototyping of other infrastructure components is moving forward using the open source Medical Device Coordination Framework [11] jointly developed by Kansas State University and University of Pennsylvania.

## References

[1] D. Arney, S. Fischmeister, J. M. Goldman, I. Lee, and R. Trausmuth. Plug-and-play for medical devices: Experiences from a case study. *Biomedical Instrumentation and Technology*, 43(4):313–317, July 2009.

[2] D. Arney, J. Goldman, S. Whitehead, and I. Lee. Synchronizing an x-ray and anesthesia machine ventilator: A medical device interoperability case study. In *Proceedings of International Conference on Biomedical Electronics and Devices (BioDevices 2009)*, Jan. 2009.

[3] ASTM Committee F-29, Anaesthetic and Respiratory Equipment, Subcommittee 21, Devices in the integrated clinical environment. Medical devices and medical systems — essential safety requirements for equipment comprising the patient-centric integrated clinical environment (ICE), 2009.

[4] C. Boettcher, R. Delong, J. Rushby, and W. Sifre. The MILS component integration approach to secure information sharing. In *27th IEEE/AIAA Digital Avionics Systems Conference (DASC 2008)*. IEEE, 2008.

[5] P. Conmy, M. Nicholson, and J. McDermid. Safety assurance contracts for integrated modular avionics. In *Proceedings of the 8th Australian Workshop on Safety Critical Systems and Software (SCS)*, 2003.

[6] R. DeLong and J. Rushby. A common criteria authoring environment supporting composition. In *Proceedings of the 8th International Common Criteria Conference*, 2007.

[7] T. Dierks and E. Rescorla. The transport layer security (TLS) protocol, version 1.2, August 2008. RFC 5246.

[8] R. Housley, W. Ford, W. P. D., and Solo. Internet X.509 public key infrastructure certificate and CRL profile, January 1999. RFC 2459.

[9] Essential safety requirements for equipment comprising the patient-centric integrated clinical environment (ICE) – part 1: General requirements and conceptual model. `http://mdpnp.org/ICE.html`, 2009.

[10] Medical device plug-and-play (MD PnP) integrated clinical environment (ICE) website. `http://mdpnp.org/ICE.html`, 2009.

[11] A. King, S. Proctor, D. Andresen, J. Hatcliff, S. Warren, W. Spees, R. Jetley, P. Jones, and S. Weininger. An open test bed for medical device integration and coordination. In *Proceedings of the 31st International Conference on Software Engineering*, 2009.

[12] Devices: General hospital and personal use devices; reclassification of medical device data system. `http://www.fda.gov/OHRMS/DOCKETS/98fr/E8-2325.pdf`.

[13] K. K. Venkatasubramanian, S. K. S. Gupta, R. P. Jetley, and P. L. Jones. Interoperable medical devices. *Pulse, IEEE*, 1(2):16–27, 2010.