computers

MDPI

Article

A Security Analysis of Cyber-Physical Systems Architecture for Healthcare

Darren Seifert and Hassan Reza *

Department of Computer Science, University of North Dakota, Grand Forks, ND 58201, USA; Darren.Seifert@gmail.com

* Correspondence: reza@aero.und.edu

Academic Editor: Kartik Gopalan

Received: 23 June 2016; Accepted: 25 October 2016; Published: 31 October 2016

Abstract: This paper surveys the available system architectures for cyber-physical systems. Several candidate architectures are examined using a series of essential qualities for cyber-physical systems for healthcare. Next, diagrams detailing the expected functionality of infusion pumps in two of the architectures are analyzed. The STRIDE Threat Model is then used to decompose each to determine possible security issues and how they can be addressed. Finally, a comparison of the major security issues in each architecture is presented to help determine which is most adaptable to meet the security needs of cyber-physical systems in healthcare.

Keywords: cyber-physical systems; software architecture; security; healthcare; blackboard; publish-and-subscribe; STRIDE

1. Introduction

Recent advances in embedded systems design, communication protocols, sensor technology, and mobile computing are enabling the development of a new class of system that integrates cyber space and our physical environment. While those working in the area of embedded system design are focused on designing computational models for specific applications, those working on these cyber-physical systems (CPS) are focused on establishing communications models that can reliably integrate time and feedback control into the model [1]. These CPS can assist us in monitoring and modifying the physical world in which we live and enhance our daily lives. Applications in aircraft and vehicle control systems, factory automation, weather forecasting, and deep sea drilling have already been identified [2].

One area where CPS could potentially provide a much needed solution is in healthcare. In the United States, there were 35 million inpatient hospital stays in the United States in 2013 [3]. In addition, the number of people over age 65 is expected to hit 72 million by 2030, up from 43 million in 2012. Many of these people will need housing in assisted living facilities and require a great number of supports [4]. In addition, the World Health Organization estimates that there are as many as 300 million people living with a disability classified as significant or higher [5]. People with significant disabilities often require assistance from caregivers or family members to perform daily tasks. These numbers show that a shift towards the creation of integrated, facility-wide health management systems is needed. CPS have the potential to enable health monitoring systems that can transition us to a more proactive and cost-effective form of healthcare that limits the need for additional man hours for support.

Although a handful of prototype systems have been developed in cooperation with healthcare providers, a facility-wide integrated CPS has yet to be created. One primary reason for this is the complexity of the system structure in healthcare environments [6]. Creating architectural models for

CPS can be quite challenging and meeting security needs of healthcare facilities is especially critical for such systems [6].

2. Related Works

2.1. Architectures for CPS in Healthcare

CPS are frequently mentioned as evolving out of the fields of embedded and real-time systems design [7]. As a result, there is a natural tendency to look at the success embedded systems have had in utilizing variations of the process control architecture as a reason that they may be adaptable as a solution for CPS in healthcare. The process control architecture has been used in many domains that are now looking to cyber-physical systems design to integrate a larger and more complex set of sensors. Two examples of this are control systems for automobiles and environmental management systems.

The CodeBlue software framework was developed by Harvard University and utilized a publish-and-subscribe routing framework in a network of wireless nodes. This framework was utilized because it fit naturally with the needs of medical applications. CodeBlue utilizes a discovery protocol so both medical devices and hand held end-user devices could determine which sensors were currently deployed in the environment. A query interface was also employed that enabled employees to request data from specific devices and to set a filter that would only transmit to an employee if the data exceeded some threshold [8]. A similar architecture was proposed by Tan et al. [7] as its similarity to how humans interact with their environment was seen as a desirable quality in CPS.

Layered architectures of several different styles were proposed. MobiHealth utilized sensor devices attached to a patient and are connected via Bluetooth to a type of handheld portable device referred to as the Mobile Base Unit (MBU). This MBU can process sensor results and relay them to a surrogate host over a cellular network. E-health applications work with the surrogate host to query any information needed rather than communicating directly with the MBU of each patient [9]. A similar approach was utilized in AlarmNet [10].

Sasi and Min [11] proposed a layered framework of services for capturing sensor data in healthcare called the Bigdata framework. This framework was made up of the component layer, the process layer, and the application layer. The component layer provides messaging and distribution services to the system level components and is responsible for routing data. The process layer filters sensor data streams to remove unwanted information, group data, and provide data processing based on a defined set of rules, lastly data is combined based on semantic knowledge. The application layer is charged with providing analytics services and event visualization. Lu and Fu proposed a similar architecture [12]. However, the base layer of their architecture was not simply a set of sensors, but rather ambient-intelligence compliant objects.

A blackboard architecture initially proposed by Winograd for tracking employee locations [13] was adapted by Wu et al. for use in a system called SensCare [14]. SensCare is a system for monitoring the activities of elderly people, using a heterogeneous collection of sensors, for the purpose of producing a semi-automatic lifelog. In this system, sensors contributed raw readings to a blackboard data system. An event controller is responsible for routing messages, as appropriate, to applications subscribed to the message. Additionally, a data preprocessor utilizes activity recognition algorithms to segment contributed data and user's annotations into a series of indexed activities.

2.2. Architectural Qualities for Healthcare CPS

As can be seen from the many different architectural styles used above, no architectural standards yet exist for the creation of CPS. However, attempting to implement any system without first demonstrating the validity of the architecture may lead to project problems and completion delays. In healthcare specific CPS, there are additional concerns that without a proper analysis of the architecture, a system may expose sensitive data about patients and staff or may fail to prevent patient harm.

However, creating architectural models for CPS can be quite challenging [6]. Wide variability in both the uses of CPS as well as the nature of the implementation environment makes it difficult to find a general architecture that will work across all cases. One place that can be used as a starting point is to determine a set of qualities inherent to all healthcare-based CPS. Candidate architectures can then be rated against these qualities to determine their potential for implementation [15,16].

One architectural quality that can be a key to the success of a CPS's design is its ability to cope with uncertainty [6]. CPS must be designed to be able to evolve and operate in unreliable environments. Uncertainty in CPS may be a result of deployment in a new and somewhat unknown environment, conflicting readings from a collection of sensors, or malfunctioning or missing network nodes.

One of the most unique challenges of some CPS systems only occurs in the system integration phase. The full nature of the heterogeneity of components and interactions inside a CPS system is not known until a designed system is fitted to the environment in which it will be installed. This is especially true in healthcare environments, as a designed system may be intended to be deployed in many different facilities.

It has been demonstrated that variability in the nature of the environment and quantities of device types that will need to be deployed drives the need for modeling and analyzing interactions among physical and computational and networking domains [17]. Work on the development of an analytics system for assisting with the long-term care of adults with special needs has demonstrated that powerful activity analysis algorithms are needed to infer useful activities from a limited set of sensor data. These activity analysis algorithms are highly dependent on not only the type of sensors used, but also the context in which they are used [18].

The scalability of a CPS's architecture is also a key to the success of the system [6]. The heterogeneous nature of the system requires flexibility in design that may need to cover physical domains including motion control, chemical and biological processes, as well as human involvement. The cyber domain will likely combine multiple types of networking infrastructure, development tools, and systems infrastructure. The ability to scale each of these components as necessary as the entirety of the system grows is crucial to its usability.

Adaptability of the system architecture is also important. Medical devices and systems will need to be dynamically reconfigured and distributed as needed so they can interact with the facilities' patients and staff when needed. Often, healthcare devices must be assembled into entirely new configurations to match specific patient or procedural needs [19]. The architecture and supporting systems of the CPS must be able to cope with high amounts of reconfiguration without undue burden on either clinicians or systems administrators.

The reactivity of the system can also be seen as an important quality in CPS for healthcare. The ability of the architecture to support rapid changing of system goals can enable operations with an immediate deadline to be processed first [7]. It is also important that a CPS for healthcare support the ability to shift goals and priorities as patients come and go or simply as their course of treatment changes. A system that is unable to readily shift these duties may become overly saturated with irrelevant data, or expose additional opportunities to attackers.

Although various CPS systems may each have an additional list of essential architectural qualities one that is integral in healthcare is security. It is crucial to patients of a facility that their data be free from compromise both by active and passive exploits. Not only must patient data not be disclosed, but it must not be able to be modified. It is also critical that any sort of disruption that may take place in the transmission of information throughout the CPS not lead to patient harm. This is important not only for patient safety and privacy, but also to the long term viability of facilities that deploy CPS. One study has found that 54% of patients were at least moderately likely to switch health providers after a security breach [20].

Additionally, security in a CPS isn't limited in scope to restricting access to traditional patient heath records. Each new type of data gathered for the purpose of ensuring patient well-being has the potential to introduce additional security issues. For example, one study working with adults with special needs

identified that in order to properly track patient activities, the collection and categorization of camera data, in addition to information from other sensors, was required [18]. The storage of these images and the resulting analysis each introduce new potential targets for attackers and the implementation of an architecture with in-built protections for evolving data types is critical.

As security is not a primary quality in any of these architectural styles an initial examination of the major styles will be performed. From this, the highest rated styles will each be adapted with an appropriate secure architectural pattern for a more thorough security examination.

3. Case Study

In order to properly explore the security issues of various candidate architectures a sample application from the problem domain is needed. The usage of smart infusion pumps was chosen for this purpose. Studies were readily available that had reviewed current pump technologies, what sort of issues staff had interacting with them, and recommendations for future improvements including desires for highly interconnected devices similar to those in a CPS.

A review of recent studies showed that smart pump technology is, in many ways, still evolving. Recent hardware revisions have augmented devices to include medication libraries that, in addition to drug names, include both soft and hard dosage limits. [21]. Drug libraries are typically custom loaded for the area of a hospital in which the device is being used. The soft and hard dosage limits are programmed by each individual facility rather than the device manufacturer [22].

However, work to provide an optimal method to deliver drug information is not complete. Studies have mentioned that while these drug libraries do help catch many potential problems, there are still sources of errors that have not been fully addressed [23]. These included clinicians overriding soft and hard limits [21] and facilities setting soft warning levels too strictly or incorrectly. In some cases this led to "alert fatigue" on the part of the nurses [24]. Drug libraries were also found to be of little use when dispensing experimental drugs or in uncommon methods for rare diseases [21]. An additional augmentation in newer pumps is the inclusion of a barcode scanner. Barcode scanning is being used to cross check prescriptions [21], but additional uses of the technology such as augmenting clinician authentication have not been addressed as of yet.

In addition to these newer smart features, many types of pumps have a set of safety features that should be accounted for in any CPS enabled system. These include audible/visible alarms when errors are detected, ports enabling connection to the nurse call relay system, and a lockable interface (with the exception of the stop button) with a single PIN code to unlock [22]. Lastly, it was found that some common practices recommended by the Food and Drug Administration, such as independent verification of high-risk drug dispensation, were not codified in existing pump technology [24].

4. Application

4.1. Architectural Quality Analysis

In order to determine which of the candidate architectures are an appropriate fit for the design of CPS for healthcare a comparison of essential architectural qualities was conducted. This analysis focused on four primary architectural styles. First among these was a process control variant called control loop. This architecture was chosen for its history of use in closely related systems such as environmental monitoring and vehicle control systems. In addition the layered architecture, publish-and-subscribe, and blackboard were all chosen due to their use in previously proposed or prototype implementations of healthcare related CPS [7,9–14]. The essential architectural qualities and a description of each in terms of CPS are detailed below:

- Heterogeneity—Diversity of devices, especially in a healthcare environment, is a key way in which many facilities overcome problems with any one manufacturer's design flaws.
- Reactivity—CPS are often deployed, both to monitor a situation as well as to react when one or more conditions change to help maintain some state or work towards a goal. For example,

CPS enabled pacemakers have the potential to allow medical professionals to ensure faster heart rate convergence to ideal values [25]. However, without architectural level support for reactivity the CPS may be unable to adapt to the change in system priorities required to promptly notify physicians and implement their instructions.

- Adaptability—Physical variations like the number of floors, how much interference will be generated, locations of rooms, employee responsibilities, etc. will all vary for each installation of a CPS.
- Scalability—the size of each deployment of a CPS will vary significantly. Naturally, as the size of a CPS is scaled up so must the ability of the system to process data in parallel.
- Uncertainty—as many CPS are unable to fully capture the circumstances under which they are operating, a software architecture that is capable of dealing with an amount of uncertainty is desirable.
- Robustness—it is particularly important in a healthcare environment to ensure that the CPS be ready to carry out its responsibilities even in light of erroneous or missing data elements. This can partially be achieved through the use of redundancy, but should also be supported through the flexibility of the system to implement sophisticated algorithms that can utilize multiple sources of data to determine which elements are erroneous.

4.1.1. Control Loop Architecture

A CPS implemented using the Control Loop Architecture would be implemented with a set of sensors that receive feedback/environmental data from the healthcare facility. Data would be fed to a control mechanism to determine how far readings deviated from their expected values. Corrections could then be fed to appropriate actuators to make adjustments to the physical environment before additional sensor readings took place. Figure 1 details a typical implementation of the Control Loop Architecture for Embedded Systems.

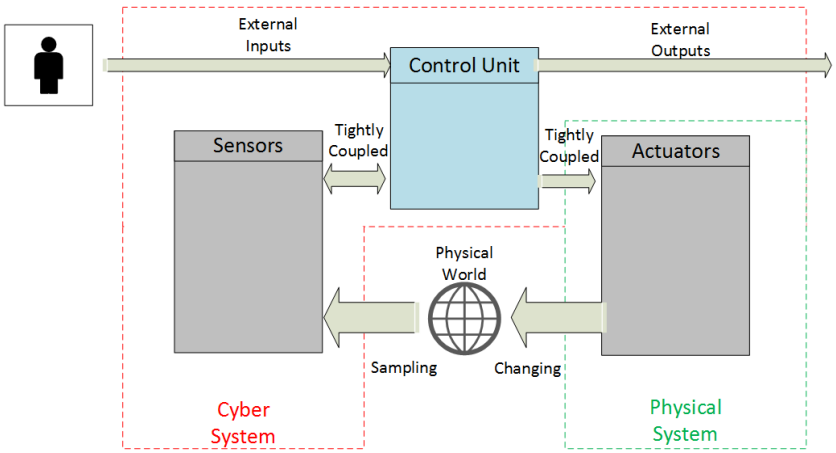

Figure 1. Control Loop Architecture for Embedded Systems (adopted from [7]).

However, the control loop architecture provides little or no guidance in how to break down complex tasks among a diverse set of sensors and actuators and appears better suited for simple tasks that require no reorganization or refinement of how devices will solve new problems. It also provides no guidance as to how to change the monitoring domain. A single control component is specified and, while it may be possible to treat this as a swappable element as goals are adjusted, little guidance exists to specify how sense/act nodes would adapt to this type change. Uncertainty is an issue with the control loop architecture as it is only able to deal with incremental corrections to a problem.

Finally, the Control Loop architecture has a modest level of support for robustness. While there is limited support for robustness through redundancy or sophisticated algorithms that can identify erroneous data in the control loop architecture the in-built mechanism for incremental corrections can

be used to gradually move the system to a safe state in spite of missing data. Unfortunately incremental corrections are not ideal in all situations.

4.1.2. Layered Architecture

The layered architecture is described taking place at two different levels in the literature. The first of these begins with the physical layer and moves up to sensors, sink nodes that collect that group related sensor readings, and control nodes that collect and group events from sink nodes. The second description takes a higher level view typically describing most of the sensor infrastructure as a single level, local event creation as a second level, and cloud infrastructure and applications that interact with it as one or more higher levels. As the purpose of this examination is to look largely at architecture of the systems local to the hospital, we will be examining the first case. Figure 2 shows one proposed implementation of a CPS using a Layered Architecture.

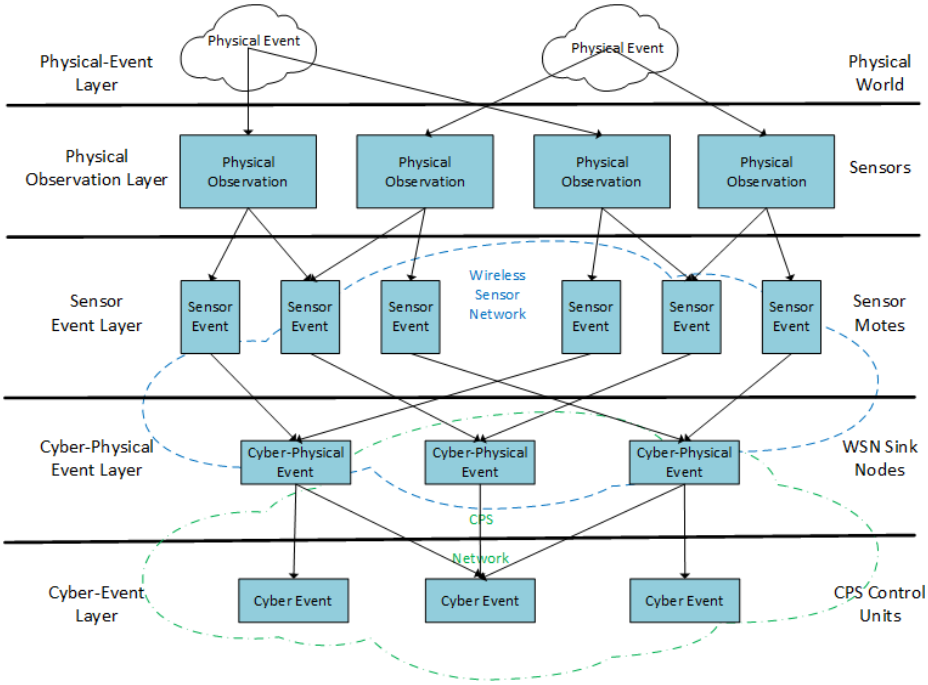

Figure 2. Layered Event Model for CPS (adopted from [26]).

Reactivity is one of the chief criticisms of layered approaches where timeliness may be a concern. Little is provided by the architecture to determine what would be done to account for actions that must be performed as a result of an event. It could be assumed that a high level action request would be sent to the level below to be repeatedly decomposed until it was broken down at the lowest level and one or more actuators were activated, but in many cases where a direct action is required this can cause delays. The same can be said for events that require immediate action, but must be sent up several more layers before an actual response is initiated.

A layered architecture will struggle with the mobility of nodes. The combination of observations before being sent to higher levels becomes difficult as sensors move through an environment. In order for this to work properly, the combining of sensor readings might have to be allowed to migrate between layers of the architecture, or the burden of processing largely local events may have to be conducted at a global level, effectively eliminating any advantages the layered architecture was supposed to provide.

The combination of observations at lower levels before being sent to higher levels typically seen in a layered architecture does offer a mechanism to build robustness into the architecture. Redundant nodes at a lower level may report data, which can then be jointly analyzed to improve confidence

levels in detected readings. These redundancies may be then be removed from the transmission stream and sent on to higher levels in the architecture where additional data streams may be combined.

4.1.3. Publish-and-Subscribe Architecture

In a publish-and-subscribe architecture, each sensor and actuator device within the network would function as an independent node. As nodes require no prior knowledge of each other additional nodes that work with employee, patient, drug databases could be added to the network as needed. Each node would then advertise and publish information as appropriate. Each sensing and actuating node would also be responsible for subscribing to information streams as appropriate. Figure 3 details one proposed implementation of the Publish-and-Subscribe Architecture for CPS.

This in-built ability of publish-and-subscribe nodes to adapt to the addition of additional node to the network provides an excellent opportunity to integrate robustness into the architecture. A node may subscribe to all data streams for a particular subject and then combine the readings through the use of algorithms. However, in practice there may be some issues with this. The limited processing ability of some nodes in the network (Implanted devices for example) may make it difficult for some nodes to handle the sophisticated algorithms required to process redundant data sources on their own. At the very least offloading the processing of these data to another publisher would add a layer of complexity not present in some other architectural candidates and in some ways violate the basic concept of the architecture.

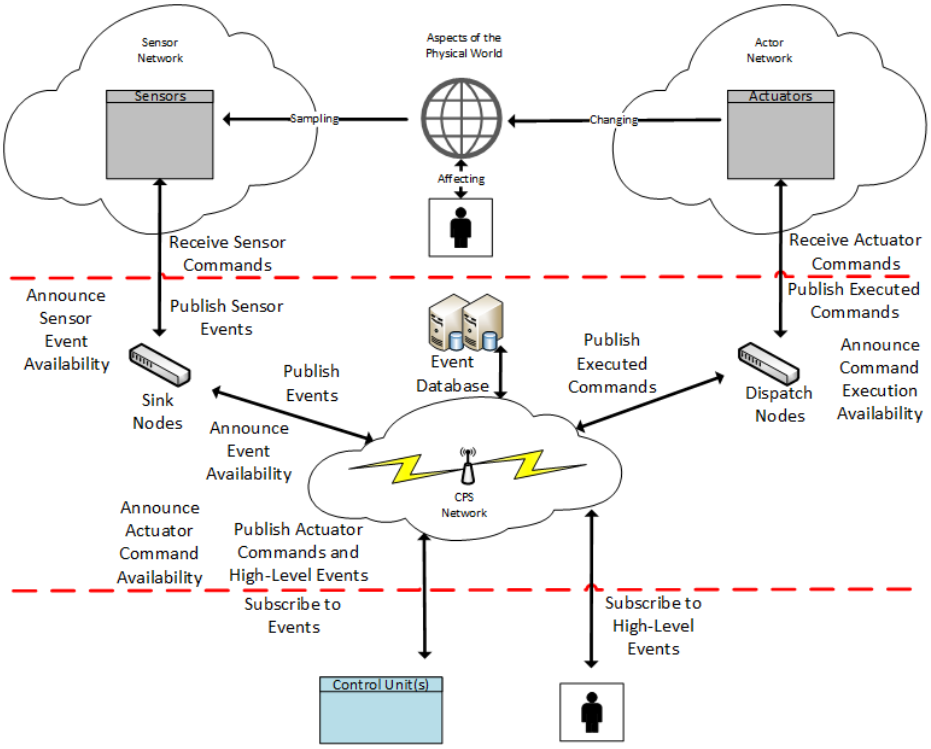

Figure 3. Publish-and-Subscribe Architecture for CPS (adopted from [7]).

Communication requirements can cause problems as the system grows. Each announcement to be broadcast can generate a lot of traffic. Likewise, the fact that every publication is sent directly to each node can result in a single sensor reading being transmitted several times. The redundant nature of sensors in a medical environment required to produce a robust system also may have an impact on this. Subscribe, unsubscribe, and unannounce messages all generate additional traffic. Uncertainty in the publish-and-subscribe architecture can be a difficult problem to cope with. The lack of any guarantee

of a producer (or conversely a subscriber) of required information can make it difficult to ensure that an algorithm is able to produce an answer for all possible situations.

4.1.4. Blackboard Architecture

The blackboard architecture would consist of independent sensing and actuating nodes much like in the publish-and-subscribe architecture. However, instead of being responsible for communicating directly with other nodes as needed, blackboard nodes or "knowledge sources" would be responsible for contributing their information to a shared data store. A control component would be charged with managing when each node could contribute information and, in many cases, when calculations on new data could begin. Figure 4 shows what an implementation of a CPS using the Blackboard Architecture may look like in a healthcare environment.

The blackboard architecture is specifically designed to work with a diverse set of knowledge sources and solve problems with uncertain solutions. Scalability is well supported, as information from knowledge sources only needs to be transmitted once to the central blackboard.

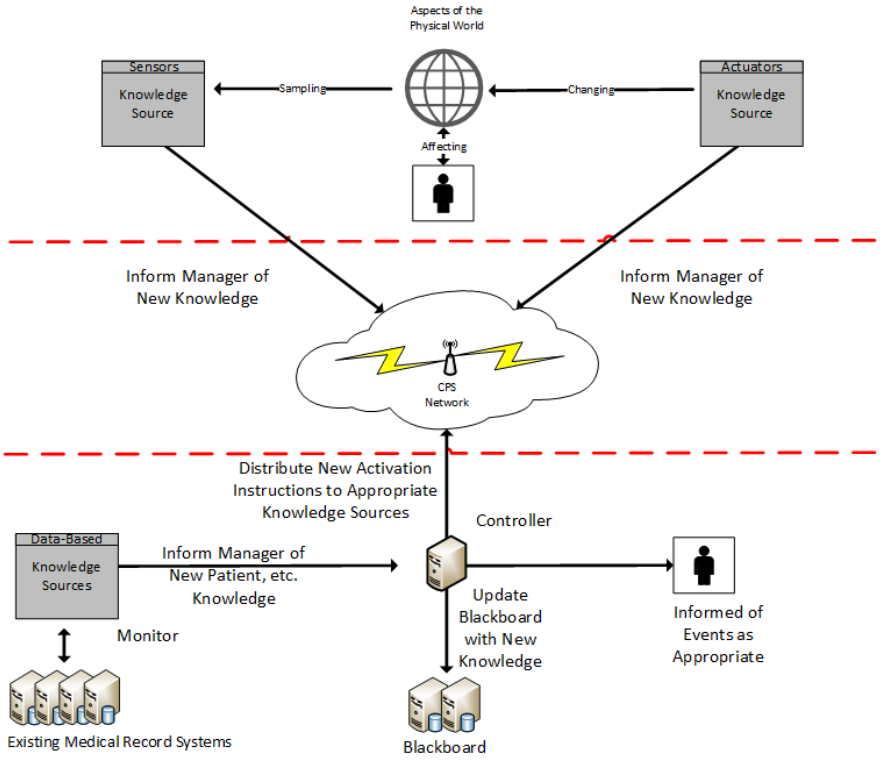

Figure 4. Blackboard Architecture for CPS.

Reactivity is built into the model through the ruleset of the control component. As information changes the ruleset allows adjustments to system priorities and knowledge source activations.

Lastly, the blackboard architecture has in-built support for robust design. The ability to have multiple knowledge sources each utilizing a variety of differing or redundant sensors and algorithms that each contribute information to the solving of a complex problem is central to the architecture. Complex decision making algorithms that can work to eliminate erroneous data can be centralized within the blackboard manager.

4.1.5. Qualitative Analysis of Candidate Architectures

Using the qualities defined above, each candidate architecture was ranked on a three point scale. A ranking of + means there is sufficient support within the architecture for the listed quality. A ranking

of + − indicates that there is some support for the quality. Lastly, a ranking of − indicates minimal support for a quality within the architecture. The results of this analysis are listed in Table 1.

Table 1. CPS Architectural Quality Comparison.

	Control Loop	Layered	Publish and Subscribe	Blackboard
Heterogeneity	−	+	+	+
Reactivity	+ −	−	+	+
Adaptability	−	−	+	+
Scalability	+ −	+	+ −	+
Uncertainty	−	+	+ −	+
Robustness	+ −	+	+ −	+

4.2. Expected Functionality of CPS-Enabled Pumps

The collection of current and future functionality desired of CPS-enabled pumps was collected and codified as required features of current pumps, desired functionality implementable due to the inclusion of CPS, and desired functionality that was not implementable.

Expressed functionality that was determined to be implementable using a CPS included independent verification of pump settings, tracking patient conditions and physician orders, additional error notifications sent directly to clinicians, addressing incorrect or missing drug labels and addressing the limitations of current drug libraries.

An analysis of these functionality requests was conducted, which led to the merging of some existing functionality with new CPS features. Out of this process a short list of expected features was developed that each required the CPS for functionality. These include:

- Clinician authentication
- Delivery of a new medication
- Medication dosage is outside of soft/hard limit
- Logging of device activity
- Alarm/shutdown for detected device error
- Dispensation of a high risk medication
- Dispensation requires vital signs data gathered by another medical device.

4.3. Adapting Architectures with Secure Patterns

As mentioned, the initial architectural analysis of candidate architectures showed that publish-and-subscribe and blackboard architectures were likely to be the two most suited for adapting for use in a healthcare CPS. However, many security issues still existed within both architectures. To combat these issues both were adapted with an appropriate security pattern. These patterns were then modified as needed to work in a CPS for healthcare.

4.3.1. The EventGuard Pattern

The publish-and-subscribe architecture is built around a series of independent nodes that require no previous knowledge of each other or their capabilities. If a node is capable of producing some type of information, it may announce this capability to other nodes in the network. Any node that is interested in this type of data may then send a subscribe request to the announcing node. When the node then has data that matches the announced data type it publishes it just to those nodes that have subscribed.

One major security problem with this generic architecture is the level of trust each node must place in both its subscribers and in fellow publishers to which it is subscribed. In a healthcare environment, a rogue node could claim to have knowledge of patient information and produce only invalid readings. Likewise, a rogue subscriber to some type of patient information could be divulging this information to third parties.

In order overcome this difficulty, the EventGuard pattern prescribes the addition of a component called the meta-service [27]. The meta-service is able to authenticate all publishers and subscribers within a network. It is charged with validating announcements as well as subscriptions, unsubscriptions, and unannouncements. A node interested in publishing some type of data must request permission to announce the data type from the meta-service. An approved announcement request is signed by the meta-service and returned to the requestor for broadcasting to the network. The meta-service's signature can be verified by all nodes. This allows interested subscribers to know not only that the identity of the announcing node has been verified, but that it is allowed to publish the specific type of information in the announcements. Subscriptions and other messages are handled the same way. The meta-service is described as replicated throughout a network as needed so a localized service is always available for nodes to communicate with.

Publications in the EventGuard pattern are signed by the node producing them and encrypted using a key provided by the meta-service. This key is only known to the publisher of the information and any interested and approved subscribers. By allowing nodes to self-sign publications using an approved encryption key, the load on the meta-service is kept to a minimum while still ensuring the meta-services approval of the transmission.

Publication topics within the EventGuard pattern are also encrypted using a one-way encryption mechanism. This mechanism is known to each node within the network and allows a publisher and a subscriber that both know their information stream of interest to encrypt the topic and produce the same result. However, when the token produced by the encryption mechanism is transmitted across the network no eavesdropping attacker can tell what is being referred to.

Lastly, the EventGuard pattern provides additional safeguards that limit DDOS attack propagation in a mesh network. If a DDOS attack prevented delivery of needed information it could lead to patient injury or death. EventGuard's guarantee of sender authenticity is combined with duplicate message detection to limit the propagation of messages sent by a malicious node from travelling beyond its neighbor nodes in most cases.

Some additional changes have been made to the EventGuard pattern to adapt it for use in CPS. The meta-service has been augmented with a role based access control mechanism (RBAC) that ensures that both advertisements and subscriptions are allowed by a healthcare device's role. This mechanism is utilized by the meta-service to determine which types of advertisements and subscriptions should be permitted by an authorized node. The inclusion of this RBAC mechanism will greatly reduce the potential for disclosure of patient data due to rogue subscriptions and publications.

To increase security of the healthcare CPS the replication operation between meta-services will be performed manually by systems administrators. Our system will consist of one meta-service that is local to the healthcare device network and a second one that is local to nodes that have access to the logging database and existing data systems such as employee, patient, device, and drug databases.

The publish-and-subscribe architecture will likely be deployed in the healthcare facility with several network segments. The first of these, referred to as the device segment, will be a private network within the facility. All CPS devices that interact with patients or employees will exist on this segment either via a wired or wireless connection, with wired being preferred. While this is a private network, it will be physically accessible by the general public. Network ports in patient rooms and wireless access points will be accessible to anyone with general access to the facility and will be secured as such. The trusted meta-service component that healthcare devices will communicate with will be segmented from this network by a firewall that only allows incoming connections to limit damage by potential attackers.

Data nodes, or nodes that are able to access information within one of the facilities existing data systems, will also be segmented from the device network by a firewall. Medical device nodes will be allowed to make connections to data nodes and vice versa. The meta-service that data nodes connect to will be separate from the one device nodes connect to. This will help limit the potential for damage

from an attacker in that if the RBAC rule set is corrupted on one meta-service, the other will continue function as expected.

Lastly, a network segment will separate data nodes from the database servers that they must connect to. While connections from the device network will be allowed to be made to data nodes, no connections will be allowed to be made from the device network segment to the database server segment and no outgoing connections will be allowed from the database server segment. Figure 5 presents a high level overview of the network design for this architecture.

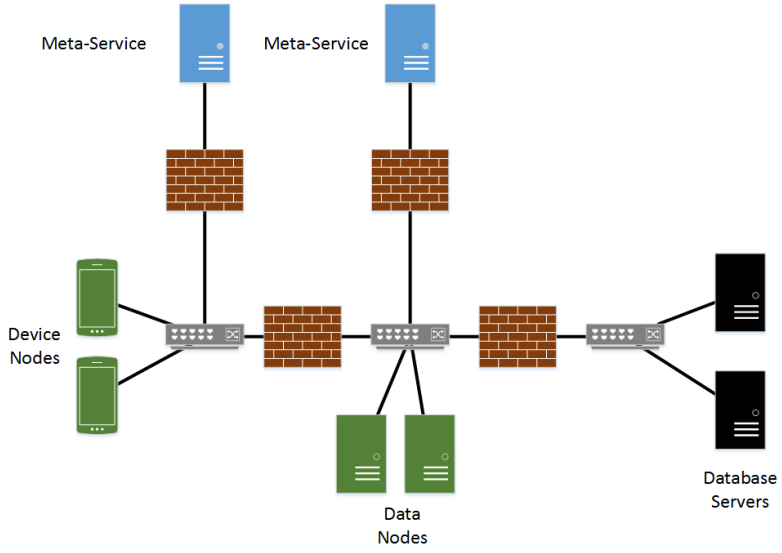

Figure 5. Publish-and-subscribe Network Diagram.

4.3.2. The Secure Blackboard Pattern

The Blackboard Architecture typically consists of a few key parts. A set of knowledge sources are capable of reading and updating data within a centralized shared data structure referred to as the blackboard. Knowledge sources may not communicate directly with each other instead placing all knowledge contributions directly into the shared blackboard. A control element is charged with monitoring the blackboard for updates and prioritizing knowledge source activations. This maps nicely on to the typical elements of a CPS with nodes with one or more sensors attached serving as knowledge sources, a database or custom data structure serving as the blackboard, and a daemon that monitors the blackboard for updates and relays information to knowledge sources. The monitoring daemon represents the control component and makes decisions on knowledge source activations based on a set of rules that are able to be modified depending on the deployment.

The major security problem with this basic architecture is there is no authentication conducted that determines whether or not a knowledge source is allowed to contribute information to the blackboard. Also lacking is some sort of mechanism to limit authorized knowledge sources to contributing or requesting information appropriate to their function.

The Secure Blackboard pattern attempts to rectify these issues by introducing three new components [28]. First among these is an authenticator that validates that the knowledge source is legitimate. Second, a reference monitor is charged with verifying the requested operation type is allowed using a RBAC mechanism. Lastly, a secure logger records the requested operation after it is performed.

The nature of CPS in healthcare required a few additional modifications to this pattern. First among these is augmenting the design to recognize the distinction between knowledge sources that work with patient devices, or device knowledge sources, and knowledge sources that work with existing data systems, or data knowledge sources.

Device knowledge sources will exist within a private network that is physically accessible to the general public much like in the publish-and-subscribe architecture. As a result, it may be possible for an unknown device to be connected to this network segment and attempt to make changes to the blackboard knowledge structure. It is also possible that, since the network segment is physically accessible, an approved device may be tampered with by an attacker. As a result, device knowledge sources will be required to connect through the control component as described in the secure blackboard pattern. As data knowledge sources will not be exposed to nearly as many types of physical attacks, they will share a separate network segment with the blackboard manger and the blackboard database and will not connect through the controller. Connections through the firewall from the controller will only be allowed to the blackboard manager and the blackboard database server. Knowledge sources within the device network segment will be unable to communicate with any component on this segment.

The control component described in the secure blackboard pattern will be divided into two components. The first of these, the blackboard manager component, will perform some of the duties typically given to the controller. It will be responsible for monitoring the blackboard database for new facts. It will also utilize a ruleset to combine facts and make determinations as to when additional knowledge source activations are appropriate. Lastly, it will take in information registrations from knowledge sources and incorporate them into the ruleset. This will leave the controller component to manage authentication.

Data knowledge sources will exist on the same segment as the blackboard manager and the blackboard database and, as they are not publicly accessible, will not be required to connect to the controller to authorize contributions to the blackboard database. However, they will be required to verify their identity to the blackboard manager and data system using an encryption mechanism. Network connections from the device segment this segment will not be allowed. Existing database servers will reside on an additional segment and communications will only be allowed to each from only the appropriate data knowledge source. Figure 6 presents a high level overview of the network design for the blackboard architecture.

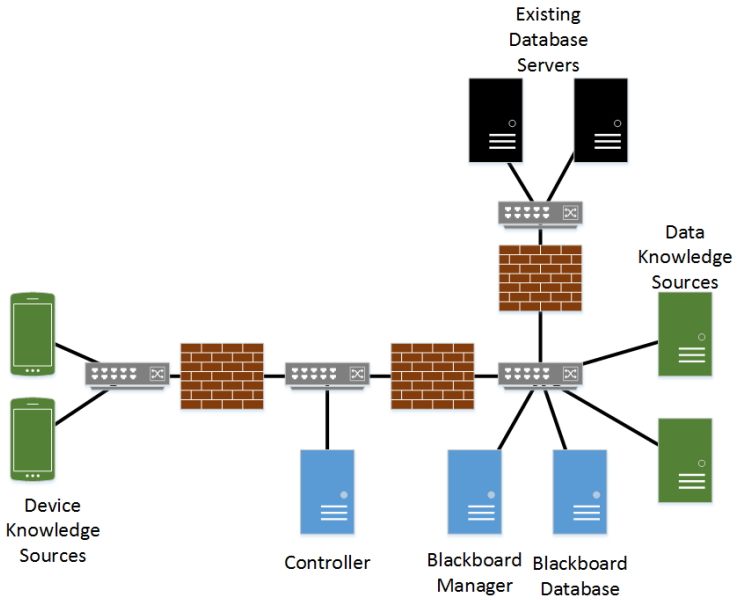

Figure 6. Blackboard Network Diagram.

4.4. Threat Modeling

In order to ensure an end-to-end analysis of the security issues presented by each architecture, a threat modeling process originally developed by Microsoft was utilized. This process was chosen both

for its structured approach and because it was specifically developed to analyze the architecture of an application [29]. Phases of Microsoft's threat modeling process include defining the boundaries of the system through the documentation of its external dependencies and identification of the assets of the system that must be protected. An architectural overview based on the expected system functionality including simple diagrams detailing trust boundaries and data flow also must be performed. Lastly, the application must be decomposed to identify attack points, privileged code elements and assets of the system.

Once the system decomposition is complete threats to the system can be identified, documented, and rated. Threat identification, categorization and documentation are performed using Microsoft's STRIDE threat analysis process. STRIDE is a goal-based approach to threat identification that utilizes the previously developed list of system assets to determine possible goals of an attacker and how they may possibly accomplished utilizing the possible attack points. Any identified attacks are then categorized as belonging to one of the six threat categories: spoofing, tampering, repudiation, information disclosure, denial of service, or elevation of privilege.

After completing the threat classification, a ranking of each threat is performed in order to distinguish threats that pose a significant risk from those that have little effect and low risk of exploitation. Risk analysis in our model was performed using a modified version of Microsoft's DREAD threat risk-ranking system. DREAD calculates risk using a number of factors including damage potential, reproducibility, exploitability, affected users, and discoverability. These individual rankings are then averaged to determine an overall score for the threat. DREAD was chosen as it is one of the most thorough mechanisms for evaluating threat risk. To improve the consistency of rankings within the DREAD model a rubric was created for each DREAD element. This rubric was used to determine each of the component scores for each threat.

This threat modeling process was conducted for each of our candidate architectures and determined the quantity, type, and severity of threats to each. These results then formed a basis on which the most suitable architecture could be chosen. The sections that follow will provide additional details about the threat modeling of the architectures and key realizations that were found when selecting the most suitable architecture.

4.4.1. Flow Diagrams

An example from each candidate architecture is presented here to show how analysis of CPS activity through the use of flow diagrams brought up some interesting points of discussion. A few of these discussion points are further detailed during the STRIDE threat analysis. Figure 7 details the first flow diagram, dispensation of a medication that requires vital signs data gathered by another medical device in the publish-and-subscribe architecture.

This flow diagram begins with the smart pump requesting authorization of the delivery of a new medication (detailed in another diagram). As part of the authorization decision, the device is informed that it is required to monitor the patient oxygen saturation levels to ensure that the dosage rate is not harming the patient. This information is sent on to the listener daemon that then listens for advertisements of oxygen saturation data about the currently assigned patient.

Next, the pulse oximeter connects to the network and requests information about which patient it is monitoring and receives a reply (detailed in another diagram). After completion of this start up process, the pulse oximeter requests a signed advertisement from the meta-service. The meta-service verifies the device's identity and then verifies that the device's role allows it to publish oxygen saturation readings. A signed advertisement is then sent back to the pulse oximeter to broadcast to all devices.

The signed advertisement is received by two nodes that recognize the encrypted advertisement token. One device is the smart pump and the other is the data node charged with logging information. Each node requests a signed subscription from its local meta-service which, in turn, verifies the identity of the nodes and then checks that the RBAC permits each node to subscribe to this type of data.

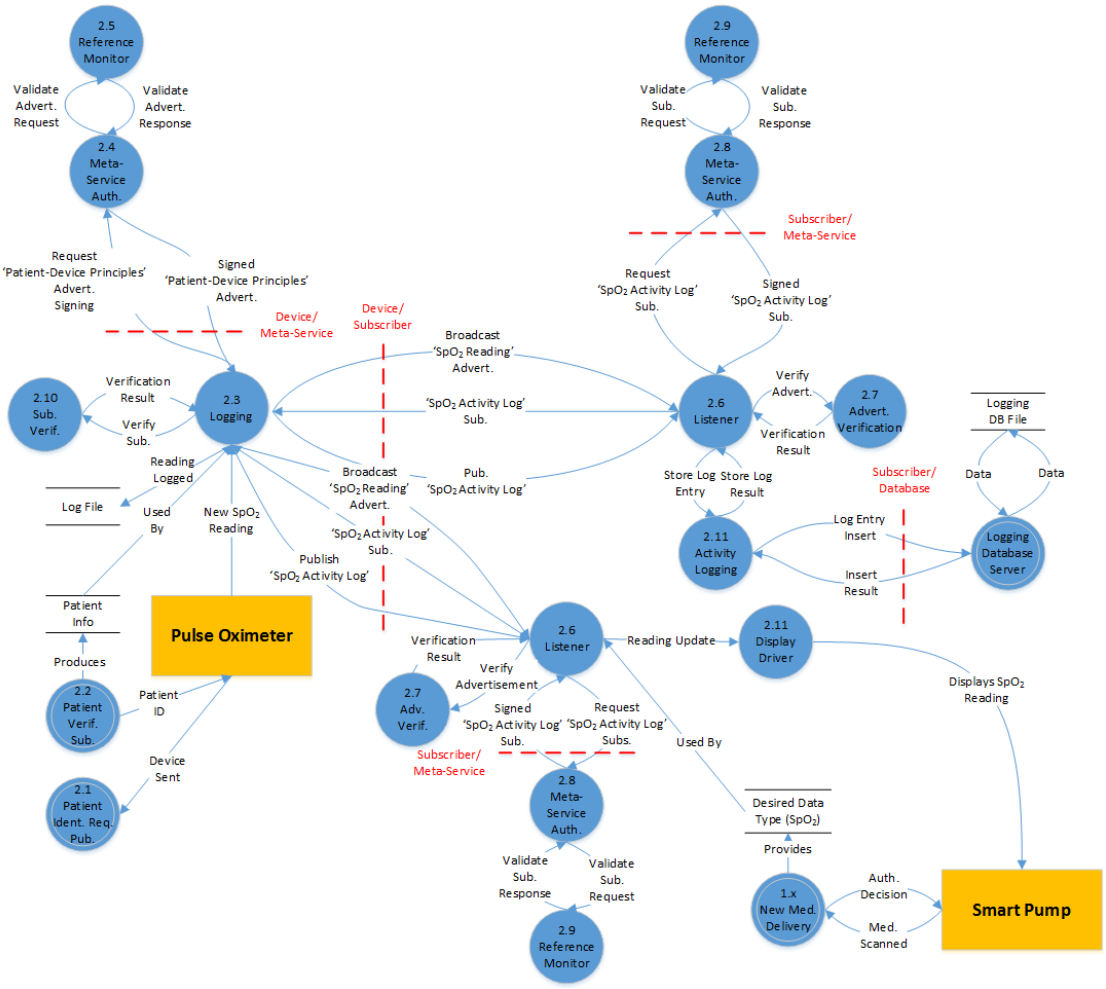

Figure 7. Drug dispensation requires data gathered by another medical device (Publish-and-subscribe).

Once each of the subscribing nodes receives their signed subscriptions they transmit them directly to the pulse oximeter, which registers their interest in its subscribers data set. When a new oxygen saturation reading is generated, the data are then signed by the pulse oximeter and encrypted using a key given to both the publisher and subscribers during the signing process. The data are then transmitted to each subscriber. Both subscribers are able to verify the readings origin and decrypt the transmission. The logging data node then logs the reading in its database and the smart pump uses the reading to regulate the rate at which it is dispensing the current drug.

One issue with this process that can be seen is the disconnect that occurs in any process that requires two-way communication. The process for the pulse oximeter to verify which user it is monitoring requires it to advertise and publish authentication principles and then a specifically crafted advertisement dedicated to that single device id has to be issued in order to publish results of the verification.

While this process by itself is not insecure it is inefficient. It can also be seen that just to complete this data flow, listening daemons are needed on both meta-services, both medical devices, the logging data node, and the logging database server. In addition, to complete the patient identification and medication verification portions of the diagram, another two data nodes and two databases must be listening for communications. While some of these ports can be protected via firewall policies, many of them must be available for any device on the physically accessible medical device network segment to connect to. As will be explained later, this led to an increased threat classification score for many exploits when compared to similar exploits in the blackboard architecture.

Lastly, and most severely, a problem exists in the RBAC and Publish-and-subscribe mechanisms for this architecture. When the smart pump is listening for oxygen saturation readings it should only be able to request readings based on the patient it is currently providing services to. Likewise, the pulse oximeter should only be able to advertise readings about the patient it is currently connected to. However, no mechanism exists to verify that this is the case.

There are possible solutions to this problem. First, the RBAC mechanism could be expanded to be able to verify patient and other types of information before signing an advertisement or subscription request. This would limit a device to only creating signing requests for a patient they are currently monitoring. It is, however, a difficult adjustment to make as the meta-service is now required to be able to verify, at a minimum, which devices are assigned to which patients. Either giving the meta-service the ability to act as a node in the network and publish verification requests or adding a direct connection from the meta-service to query the appropriate database server would help to overcome this portion of the problem. However, it does nothing to stop a device from advertising readings for a patient it was at one time assigned to as the meta-service has no advertisement or subscription revocation mechanism.

A second solution might be to allow the smart pump to verify for itself that the pulse oximeter is indeed currently assigned to the patient that it is advertising readings about. However, this opens up an additional security problem. When examining this solution from a different perspective, it can be seen that this would probably require that each device have permission to verify which patient every other device in the network is assigned to. This could become an exploitable mechanism for wide scale information disclosure.

To contrast this analysis of the Publish-and-Subscribe Architecture, a diagram was chosen of the dispensation of a high risk medication in the Blackboard Architecture. This flow diagram, detailed in Figure 8, begins with validation of the device to the controller. Communications from the devices to the controller maintain a persistent connection in the Blackboard Architecture. This allows the controller to relay information to each device as needed without having a listening port open on each device. When the device is first powered up and communications are established, the device connects to the controller to verify its identity and establish its encrypted data stream.

Once the first clinician is ready to dispense the high-risk medication they proceed through the user authentication process that begins with them scanning their staff id card and entering their individual user passcode into the pump. The clinician then uses the barcode scanner to scan the new medication. This process provides the pump with two pieces of information. The first is the identity of the patient that the drug is assigned to and the second is the type of drug and its prescribed dosage. The communications management component then relays to the controller that it has knowledge of a medication scanning event and includes the scanned patient id and the medication information.

Once the controller receives this update from the device knowledge source, the reference monitor verifies that this device is allowed to contribute knowledge of new medications using RBAC. If this is found to be a valid role of the device, the controller commits the information to the blackboard database.

The blackboard monitor, charged with monitoring updates to the database, sees the addition and, utilizing its ruleset, makes the decision to activate the medical device knowledge source and inform it of the new fact. The medical device knowledge source specifically needs information from the record on the contributing device and the patient recorded from the scanned barcode. At this time, the blackboard monitor also utilizes its ruleset to decide to activate the medication knowledge source. The medication knowledge source is sent the patient ID, medication name, and the prescribed dosage rate. Upon receiving the patient and device information from the blackboard monitor, the medical device knowledge source issues a query to the medical device database to verify that this device is currently assigned to the patient. The knowledge source then contributes this verification information to the blackboard database.

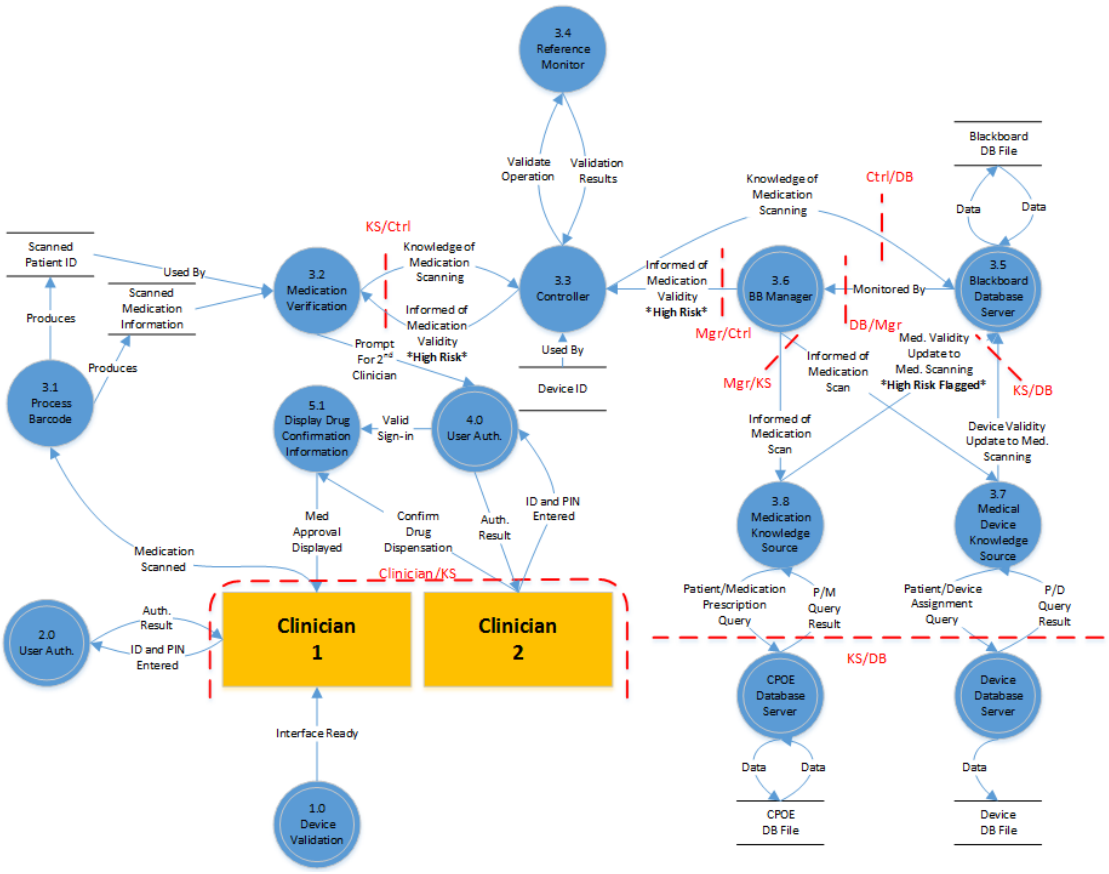

Figure 8. Dispensation of high risk medications (Blackboard).

When the medication knowledge source is activated, it has two responsibilities. The first of these is the verification against the customer provided order entry (CPOE) database that the patient was indeed prescribed the medication at the prescribed dosage. The second duty is the verification against the drug library that the prescribed dosage is within the proper limits. As part of the medication verification process, it is noted that this particular drug has a high-risk for patient injury. The knowledge source contributes this medication information to the blackboard database.

Upon seeing the contributions of these two new facts, the blackboard manager utilizes its ruleset to combine the information with the original validation request and determines that it should activate the device knowledge source and inform it that the medication and patient information are correct, but that the medication is a high-risk medication. This information is sent to the controller as the blackboard manager does not communicate directly with device knowledge sources.

Once the activation notification is received from the blackboard manager the controller again utilizes RBAC to verify that the device is allowed to receive this type of information. The activation information is then passed on to the device. The knowledge source analyses the received information and, noticing that the medication was flagged as high risk, begins the process of validating a second clinician and informing them of the requested dosage information. If the second clinician agrees with the settings, dispensation of the drug may begin.

One of the first things that may be noticed with this architecture is two-way communication processes such as user authentication are a single node in the flow diagram rather than two separate nodes as in the publish-and-subscribe architecture. The way knowledge sources pass messages through the central blackboard can make it seem like there may be a similar disconnect with two-way communications to that of publish-and-subscriber architecture. However, there are some key differences. In the blackboard architecture there would be no inherent connection between contributing the medication verification request and receiving the validation within the device.

However, the architecture does provide a guarantee of a single component existing that can both listen to the request and provide an activation notification when the medication validation fact is resolved, whether that be by another knowledge source or through some sort of ruleset that combines existing facts and algorithms to come to a conclusion.

Contrasting this with the publish-and-subscribe architecture, a single node can advertise information and wish to subscribe to publication streams, but there is no guarantee that another node will even subscribe or publish as needed. This can make development of a comprehensive implementation more difficult as a higher level of coordination is required between all device manufacturers as well as the supplier of the meta-service software.

Unlike in the publish-and-subscribe architecture, there are only a few systems that must be listening for network communications. In this flow diagram the controller, the blackboard manager and the three database servers are all that must have a component that listens for network communications. Knowledge sources all establish persistent outgoing connections to the hosts they communicate with to limit the network profile of the architecture. Additionally, only the controller listens for connections from the healthcare device network segment.

This limited attack profile has its advantages and disadvantages. First it is easier for network administrators to monitor. However, at the same time it gives attackers a single target at which to attempt DDOS attacks. A successful attack of this type may interrupt communications for the entire CPS.

This flow diagram also shows how the RBAC system works much better within the blackboard architecture. As there are no communications allowed directly from one knowledge source to another, there is no need for them to trust one another. They only need to trust the identity of the controller. The information initially contributed by the smart pump contained the scanned information from the medication, which was stored in the blackboard database. This information was then verified by information directly from the hospital's databases. Any device that needed to rely on this information would know that it had been verified by administrative systems rather than self-reported information from the device.

4.4.2. STRIDE Threat Model

After decomposing the flow diagrams into a collection of external dependencies, entry points and assets an analysis of the system using the STRIDE threat model was performed. Each attack that is identified from the decomposition is placed into one of the six STRIDE categories based on what type of attack it is and what control mechanisms the architecture has in place to prevent each of them [30]. These categories include:

- Spoofing—An attacker utilizes illegally accessed credentials of some valid user to attack a system.
- Tampering—Malicious modification of data within a system. This may happen in transport, or it may involve stored data.
- Repudiation—An attack on the system that involves executing some action and then being able to deny being the source of the attack. This includes exploits that are able to be executed anonymously.
- Information disclosure—An especially dangerous type of attack in medical systems. It involves allowing an individual the ability to read sensitive information without permission.
- Denial of service—An attacker is able to make critical components of the architecture unavailable to valid users for some amount of time.
- Elevation of privilege—Utilizing a valid user account an attacker is able to attain administrative credentials or any similar exploit (An anonymous user attaining permissions of a regular user for example).

The grouping and descriptive information provided by the STRIDE model assist in making sure a thorough evaluation of the system has been conducted. However, it does not provide any

evaluation as to the significance of each attack. As a result, once the list of threats has been compiled, a separate analysis of what level of harm each threat poses to staff, patients, or the facility itself must be conducted.

As this analysis is focused on the architectural level, we have chosen to use a modified version of the DREAD threat-risk ranking model. The DREAD model ranks each threat in five different areas and then produces an overall ranking based on the scores. The original DREAD model's 10 point scale has been replaced with a four level ranking system and, prior to conducting rankings, a set of criteria were established that had to be met for a level to be chosen. The DREAD analysis is based on determining answers to the following questions for each attack:

- Damage: How much damage would result?
- Reproducibility: How difficult is the execution?
- Exploitability: What is needed to execute the attack?
- Affected users: How many people would likely be affected?
- Discoverability: How hard is it to find the vulnerability?

A small sample of the STRIDE analysis breakdown and the DREAD risk ranking for one issue discussed in the analysis of the flow diagrams for the publish-and-subscribe architecture can be seen in Table 2. A similar breakdown for the blackboard architecture can be seen in Table 3. In both tables, column one contains the basic attack information, assets that may be targeted by the attack, and a rough sketch of the technique to be used by the attacker. This information is used in the development of the description of the countermeasures in place to defend against the attack and reasoning about the possible effects of a successful execution of the attack. These items appear in column two. The last column details the DREAD risk ranking for the attack.

Table 2. STRIDE risk analysis and DREAD risk ranking for the publish-and-subscribe architecture.

STRIDE Threat List			
Attack	**Control/Reasoning**	**DREAD Ranking**	
	Spoofing		
An attacker compromises a device and utilizes previously authorized advertisements to publish information about patients the device is no longer assigned to. This may be a threat to patient data validity, but may also cause patient harm if spoofed data is relied upon by another medical device.	Compromising a device and obtaining the list of keys and signatures for its authorized advertisements is not easy and most likely would require disassembly of the device. However, flaws in the listener daemon could be utilized as part of an attack well. There is no mechanism to revoke approved publication rights either from the meta-service level, or by a more knowledgeable node. For example, the node that can communicate directly with the device assignment database likely will have the most up to date information on which patients a node should publish information about, yet has no ability to revoke a device's approved publication signing rights about any patient it's ever been approved to provide information for. Most likely it would be up to individual device manufacturers to issue unadvertise requests for previous patients when a medication for a new patient is scanned and approved. This can be a dangerous and unreliable way to implement protections.	Damage: Reproducibility: Exploitability: Affected Users: Discoverability: Overall:	Critical Important Important Important Critical Important

Table 3. STRIDE risk analysis and DREAD risk ranking for the blackboard architecture.

STRIDE Threat List		
Attack	**Control/Reasoning**	**DREAD Ranking**
Denial of Service		
An attacker may attempt to deny service to health devices using a distributed denial of services attack. A successful attack could interrupt communications between devices and the blackboard database for an extended period. No permanent damage to data is likely, but any device that cannot function independently of CPS communication may fail and lead to patient injury.	This likely will be difficult to execute as both the physical and wireless public networks are closed systems without internet access. The likelihood of an attacker being able to introduce enough devices to the network to deny connections is unlikely. Techniques such disabling unused network ports, and limiting both the number of devices and which MAC addresses can be added to switching tables for networking equipment can help to limit this. Wireless network access can be restricted using a password-based encryption mechanism as well. Some networking equipment can not only restrict this access, but can generate a security violation if a device count/disallowed MAC is connected which can be used to alert network administrators.	Damage: Medium Reproducibility: Important Exploitability: Medium Affected Users: Critical Discoverability: Critical Overall: Important

5. Conclusions

In order to make a determination of which architecture was the most appropriate from a security perspective candidate architectures were judged primarily on the number of security issues that had a DREAD ranking of important or higher.

5.1. Publish-and-Subscribe Architecture

Important issues in the publish-and-subscribe architecture revolved around several core problems. First among these was the inability of the architecture to verify device to patient, device to clinician or device to medication associations in any meaningful way. The meta-service component of the architecture has the ability to verify whether or not a particular device is able to request permission to advertise or subscribe to a particular type of data. For example, in Figure 3 the meta-service verifies that the smart pump is authorized to request blood oxygen saturation (SpO_2) readings. However, the meta-service has no mechanism for verifying which patient the device is currently monitoring. The end result of this is that the smart pump is allowed to request SpO_2 readings on whichever patient it submits the request for. If publishers and subscribers can always be trusted this is not an issue, but as they reside on a network segment that is physically accessible by the public this is not the case.

A change could be made to the architecture to enable the meta-service to verify device to patient assignments before permitting a subscription request, but that changes the nature of the architecture enough that it may not be worth pursuing publish-and-subscribe anymore. Something like the client server architecture or even a layered architecture may become more appropriate.

The second core security issue revealed with the publish-and-subscribe architecture is that there are no publication or subscription revocation mechanisms accessible to the meta-service or other outside trusted entities. If a smart pump device node wishes to publish information about its drug dispensations, possibly for inclusion in the site wide event log, it must seek permission from the meta-service. Once the meta-service authorizes the advertisement, the device node is free to publish about the topic until it chooses to broadcast an unadvertisement. Subscribers to this publication are free to come and go as long as they have received permission from the meta-service. However, if a change

is made to the device assignment data system that indicates the smart pump is no longer assigned to a patient, or it is now assigned to a different patient, the pump is still free to publish information about the previous patient. This is even true if the device requests advertisement permissions for a newly assigned patient.

Issues were also found with the architecture in allowing physician's to confirm soft-limit dosage overrides from mobile handheld devices. The distributed nature of the publish-and-subscribe architecture make it difficult to create a secure entry point in the network so devices connecting from the internet can access the meta-service and publish directly to/subscribe from device and data nodes.

To address this issue, an additional component was added to the architecture to handle all communications between devices connecting over the internet and the healthcare device network. This gateway device, referred to the mobile client manager, would exist in a DMZ between the internet and the device network. Communications between mobile devices and the mobile client manager would be encrypted using TLS, so eavesdropping on communications would be difficult.

The problem that arises is that the key-exchange based authentication mechanism that normally occurs between the meta-service and nodes would now be occurring between the meta-service and the mobile client manager. A physician's authentication attempt would be sent from the mobile device to the mobile client manager, which would then publish the request and receive verification back from the employee data node.

While this arrangement does limit the exposure of devices on the network to the internet, the ability of the meta-service to independently verify the identity of each device seeking advertisement or subscription permission is lost. This independent verification is replaced with an implicit trust between the meta-service and the mobile client manager, which can be dangerous considering the mobile client manager serves as a gateway between all internet-connected devices and the private device network.

Although encryption is employed at every level of the publish-and-subscribe architecture, indirect monitoring of patient information is a real concern. An attacker that is able to gain access to monitor communications within the private device network segment would have access to, at a minimum, knowledge of exactly which devices were communicating with each other and at what intervals. In some cases this may be enough information to lead to a disclosure of patient information. For example, using our use case from Figure 3 an attacker monitoring network traffic would likely be able to tell that a patient in some specific room had one device communicating directly with another at a set interval. Using the MAC addresses of the devices may enable the attacker to determine manufacturers and, in some cases, specific device types.

Denial of service attacks are also an issue in the publish-and-subscribe architecture. The decentralized nature of intra-node communications in this architecture combined with the limited access allowed to the health device network segment make it unlikely that someone could successfully execute a denial of service attack that resulted in a total shutdown of the CPS. However, successfully denying service to a specific device, or all devices of a single patient is extremely likely to succeed. Each device in the publish-and-subscribe architecture is required to run a listening daemon that monitors for announcements, publications, subscriptions and other messages. As many of these devices are designed to minimize their physical footprint to increase portability and patient comfort they often have a limited processing capability and may be ill-equipped to cope with a large number of simultaneous connections. CPS healthcare devices are likely to be provided by a number of different manufacturers, each with their own software development methods, which also increases the likelihood that at least some subset of devices are vulnerable to DDOS attacks. Limiting which devices have access to utilize the healthcare network can help to limit this vulnerability, but the implementation of this protection is up to the network administrators of each individual facility.

5.2. Blackboard Architecture

Many of the issues encountered with the publish-and-subscribe architecture were non-issues in the blackboard architecture due to its underlying structure. Much like the meta-service, the controller's RBAC mechanism is also unable to verify that a given device knowledge source is able to contribute information about a specific patient. However, this is of no concern in the architecture as each knowledge source is only required to contribute its individual knowledge. For example, in the blackboard architecture when the smart pump requires SpO_2 readings from another medical device it simply registers this desire with the blackboard manager, which updates its ruleset to include that information. The smart pump knowledge source doesn't need specify a specific patient as a part of this registration as patient assignments are contributed to the blackboard by the medical device knowledge source. This architectural feature enables the RBAC mechanism to limit any one device to only being able to acquire information about the specific patients, clinicians, and drugs that the device is currently authorized to work with.

Additionally, revocation of rights within the blackboard architecture is a nonissue. If a change is made to the medical device database disassociating a patient from a medical device, the knowledge source monitoring the database will contribute this knowledge to the blackboard server. A rule in the blackboard manager's ruleset will activate the appropriate medical device knowledge source and notify it of the change. If it is assumed that the knowledge source is controlled by an attacker and they attempt to continue to contribute information, the communications will be allowed by the controller's RBAC mechanism. However, as the device's assignment change is already known to the blackboard database, the blackboard manager's ruleset will prevent these new contributions from being assigned to a patient.

As the controller already serves as a gateway between an assumedly untrustworthy device network and contributions to the blackboard database it can also be used to authenticate handheld mobile devices. This allows individual physician's devices to be authenticated as any medical device would. The controller's key exchange identification and encryption mechanism would require an attacker to not only know a physician's username and password to access the system, but they would also have to obtain access to the physician's handheld mobile device.

Like in the publish-and-subscribe architecture, indirect monitoring of encrypted communications is a concern for disclosure of patient information in the blackboard architecture. However, as all communications are relayed through the controller rather than directly from device to device it is much more difficult for an attacker to identify exactly which devices are interrelated. This provides some level of protection even if a device manufacturer is unable to implement some sort of protection such as time delayed communications.

The largest security issue in the blackboard architecture is possibly its exposure to a denial of service attack. Although devices do not have to run a listening daemon in the blackboard architecture, all device knowledge source activations and knowledge contributions are sent through the controller. If an attacker was able to deny access to the controller it would disrupt the entire CPS. However, to handle the large volume of device communications, the controller will already consist of a highly available computer cluster. As a result, it is unlikely that an attacker could add or compromise enough health devices on the private network to deny services. In order to enable outside access to clinician's handheld mobile devices there will have to be an internet-accessible interface to the controller, which opens up a much more serious attack front. One solution to this is to utilize a different set of physical servers to process controller communications from the internet and use a shared ruleset between the two controller components. This would enable the internal controller component to keep functioning for healthcare devices even if a denial of service attack manages to block controller access for physician's devices.

5.3. Discussion

Due to the common domain of our two candidate architectures, many attacks were of a similar nature between the two. These attacks often only had minor differences in how they execution would take place, but had similar targets or exploitations. While many of the attacks were found to have medium or lower levels of risk in both architectures, there were some security differences noted among them.

First among these was a set of attacks that would normally require disassembly of a medical device in order to get access to some assets. These assets might include a set of encryption keys, a device identifier, a list of subscriptions or advertisements or some piece of information related to a patient, clinician or medication. Each of these attacks was determined to also be exploitable over the network in the publish-and-subscribe architecture due to the listening daemon that runs on each device. The increased attack vector raised the DREAD exploitability ranking by one level in many cases.

A much smaller set of exploits had a slightly higher ranking in the damage category of the DREAD analysis of the blackboard architecture. These exploits were primarily aimed at corrupting or altering the event log of the architecture. Logging in the blackboard architecture is a natural part of the blackboard database, which is central to the functioning of the CPS and will contain an abundance of additional health related information, whereas in the publish-and-subscribe architecture, the logging data node and database server have no additional purposes within the CPS.

Attacks that had an important or higher rating were more common in the publish-and-subscribe architecture and primarily occurred in the spoofing and tampering domains of the STRIDE analysis. The addition of the trusted third-party meta-service component through which information exchanges can be brokered does a lot to augment the abilities of the architecture, but ultimately does not produce a completely viable solution. Some issues, such as meta-service device to patient verifications, may be able to be overcome through the use of additional communication layers. However, these ultimately contort the nature of the architecture. Additionally, the inability of the architecture to provide a high-level mechanism to enforce cancellation of publication rights is impossible to overcome without a complete architectural redesign. These two issues were the root cause of many of the additional security concerns.

The blackboard architecture appeared to have no major architectural issues that prevented implementation of a facility wide healthcare CPS. Issues that were found to have an important or higher rating were limited to the possibility of information disclosure through indirect observation of encrypted communications and a denial of service attack against the control component of the architecture. While each of these is certainly a concern, they are both, in some ways, less of a concern than similar attacks in the publish-and-subscribe architecture. For these reasons, the blackboard architecture appeared to be the more appropriate choice for implementation within a healthcare CPS. A breakdown of the number of important or higher security concerns for each candidate architecture and their STRIDE classification category can be seen in Table 4.

Table 4. STRIDE Threat Model Analysis.

	Publish and Subscribe	**Blackboard**
Spoofing	3	0
Tampering	3	0
Repudiation	0	0
Information Disclosure	1	1
Denial of Service	2	1
Elevation of Privilege	0	0

Analysis of the blackboard architecture augmented with the secure pattern shows that it still meets the initial list of architectural requirements as well. The control component is specifically designed to work with a diverse set of components from multiple manufacturers while utilizing RBAC to

minimize configuration requirements. A heterogeneous collection of devices may still utilize a standard communications mechanism for relaying information to the central control component. However, encryption and authentication mechanisms are in place to protect the system from unauthorized contributions. The reactive nature of the control component's ruleset in the standard blackboard architecture is preserved in the capabilities of the blackboard manager component. Its fact-based ruleset allows the system to change calculation priorities and knowledge source activations based on the currently available information set. Lastly, both the control component and the blackboard manager have the ability to divide responsibilities across a number of sub-systems as the size of the facility and number of knowledge sources increase. The security augmentations offered by the secure blackboard pattern do little to void any advantages offered in the initial analysis of the blackboard architecture.

5.4. Future Work

This work by no means is complete. Therefore, it can be extended to an architectural analysis utilizing additional types of medical devices. Implanted medical devices such as pacemakers and artificial pancreases and complex medical equipment such as CT scanners and surgical robotics each may require a different set of actors and interactions with a CPS. Ensuring that the architecture is able to work with a wide variety of equipment types while maintaining the security of patient information is crucial to the eventual implementation of such a system.

Acknowledgments: This material is based upon work supported by the National Science Foundation/EPSCoR Award No. IIA-1355466 and the State of North Dakota.

Author Contributions: Darren Seifert and Hassan Reza identified the key architectural requirements such as security and safety relevant in EHealth-Based CPS in this work. Darren Seifert carried out the experiments and quality analysis based on STRIDE; Darren Seifert and Hassan Reza wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Marculescu, R.; Bogdan, P. Cyberphysical Systems: Workload Modeling and Design Optimization. *IEEE Design Test Comput.* **2011**, *28*, 78–87. [CrossRef]
2. Mohamed, N.; Al-Jaroodi, J. A survey on service-oriented middleware for wireless sensor networks. *Serv. Oriented Comput. Appl.* **2011**, *5*, 71–85. [CrossRef]
3. Agency for Healthcare Research and Quality. *HCUP Fact Facts*; Agency for Healthcare Research and Quality: Rockville, MD, USA, 2015.
4. Ortman, J.; Velkoff, V. *An Aging Nation: The Older Population in the United States*; US Census Bureau: Washington, DC, USA, 2014; p. 6.
5. World Health Organization. *World Report on Disability*; World Health Organization: Geneva, Switzerland, 2011.
6. Hague, S.; Aziz, S.; Rahman, M. Review of Cyber-Physical System in Healthcare. *Int. J. Distrib. Sens. Netw.* **2014**. [CrossRef]
7. Tan, Y.; Goddard, S.; Pérez, L.C. *A Prototype Architecture for Cyber-Physical Systems*; ACM SIGBED Review: New York, NY, USA, 2008; Volume 5.
8. Shnayder, V.; Chen, B.; Lorincz, K.; Jones, T.R.F.F.; Welsh, M. Sensor Networks for Medical Care. In Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems, San Diego, CA, USA, 2–4 November 2005.
9. Konstantas, D. Continuous Monitoring of Vital Constants for Mobile Users: The MobiHealth Approach. In Proceedings of the 25th International Conference of the IEEE Engineering and Medicine and Biology Society, Cancún, Mexico, 17–21 September 2003.
10. Wood, A.D.; Stankovic, J.A.; Virone, G.; Selavo, L.; Selavo, L.; Cao, Q.; Doan, T.; Wu, Y.; Fang, L.; Stoleru, R. Context-aware wireless sensor networks for assisted living and residential monitoring. *Netw. IEEE* **2008**, *22*, 26–33. [CrossRef]
11. Sasi, D.; Min, D. Medical Cyber Physical Systems and Bigdata Platforms. In Proceedings of the Medical Cyber Physical Systems Workshop, Philadelphia, PA, USA, 8 April 2013.

12. Lu, C.; Fu, L. Robust location-aware activity recognition using wireless sensor network in an attentive home. *IEEE Trans. Autom. Sci. Eng.* **2009**, *6*, 598–609.
13. Winograd, T. Architectures for Context. *Hum. Comput. Interact.* **2001**, *16*, 401–419. [CrossRef]
14. Wu, P.; Peng, H.-K.; Zhu, J.; Zhang, Y. Senscare: Semi-Automatic Activity Summarization System for Elderly Care. In *Mobile Computing, Applications, and Services*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 1–19.
15. Kim, J.; Mosse, D. *Generic Framework for Design, Modeling and Simulation of Cyber Physical Systems*; ACM SIGBED Review: New York, NY, USA, 2008.
16. Derler, P.; Lee, E.A.; Vincentelli, A.S. Modeling cyber-physical systems. *Proc. IEEE* **2012**, *100*, 13–28. [CrossRef]
17. Sztipanovits, J.; Koutsoukos, X.; Karsai, G.; Kottenstette, N.; Antsaklis, P.; Gupta, V.; Goodwine, B.; Baras, J.; Wang, S. Toward a science of cyber-physical system integration. *Proc. IEEE* **2012**, *100*, 29–44. [CrossRef]
18. Wolf, M.; van der Schaar, M.; Kim, H.; Xu, J. Caring Analytics for Adults with Special Needs. *IEEE Des. Test* **2015**, *32*, 35–44. [CrossRef]
19. Radhakisan, B.; Gill, H. Cyber-physical systems. *Impact Control Technol.* **2011**, *12*, 161–166.
20. Loria, G. HIPAA Breaches: Minimizing Risks and Patient Fears. Available online: http://www.softwareadvice.com/medical/industryview/hipaa-breaches-report-2015/ (accessed on 12 June 2016).
21. Ohashi, K.; Dalleur, O.; Dykes, P.C.; Bates, D.W. Benefits and risks of using smart pumps to reduce medication error rates: A systematic review. *Drug Saf.* **2014**, *37*, 1011–1020. [CrossRef] [PubMed]
22. Fields, M.; Peterman, J. Intravenous Medication Safety System Averts High-risk Medication Errors and Provides Actionable Data. *Nurs. Adm. Q.* **2005**, *29*, 78–87. [CrossRef] [PubMed]
23. Trbovich, P.L.; Pinkney, S.; Cafazzo, J.A.; Easty, A.C. The impact of traditional and smart pump infusion technology on nurse medication administration performance in a simulated inpatient unit. *Qual. Saf. Health Care* **2010**, *19*, 430–434. [CrossRef] [PubMed]
24. *Infusion Pump Risk Reduction Strategies for Clinicians*; Food and Drug Administration: Silver Spring, MD, USA, 2015. Available online: http://www.fda.gov/MedicalDevices/ProductsandMedicalProcedures/GeneralHospitalDevicesandSupplies/InfusionPumps/ucm205406.htm (accessed on 6 June 2016).
25. Bogdan, P.; Jain, S.; Marculescu, R. Pacemaker control of Heart Rate Variability: A Cyber Physical System Perspective. *ACM Trans. Embed. Comput. Syst.* **2013**, *12*. [CrossRef]
26. Tan, Y.; Vuran, M.C.; Goddard, S. Spatio-Temporal Event Model for Cyber-Physical Systems. In Proceedings of the IEEE International Conference on Distributed Computing Systems Workshops, Montreal, QC, Canada, 22–26 June 2009; pp. 44–50.
27. Mudhakar, S.; Liu, L. Securing Publish-Subscribe Overlay Services with Eventguard. In Proceedings of the 12th ACM Conference on Computer and Communications Security, Alexandria, VA, USA, 7–10 November 2005; ACM: New York, NY, USA, 2005; pp. 289–298.
28. Ortega-Arjona, J.; Fernandez, E. The Secure Blackboard Pattern. In Proceedings of the 15th Conference on Pattern Languages of Programs, Bavaria, Germany, 7–11 July 2010; ACM: New York, NY, USA, 2008; pp. 22–30.
29. Meier, J.D.; Mackman, A.; Dunner, M.; Vasireddy, S.; Escamilla, R.; Murukan, A. *Improving Web Application Security: Threats and Countermeasures*; Microsoft Press: Redmond, DC, USA, 2003; pp. 45–66.
30. Shostack, A. Experiences Threat Modeling at Microsoft. In *Modeling Security Workshop*; Lancaster University: Lancaster, UK, 2008.