A Cyber Model for Privacy by Design (PbD)

Authors: Mike.Davis.SD@gmail.com, Dr. Ulrich Lang, and Sid Shetye.

What do we mean by Privacy?

Does privacy protection matter in the internet of things (IoT) - including consumer electronics (CE)? So what is privacy, how is it valued and where does it sit in your organization today? Chances are if you don’t have a chief privacy officer or data protection officer, your company is lacking in protecting critical data, let alone all the laws and statutory regulations dealing with privacy (e.g., audit, compliance, etc.). Especially considering the key mandated privacy requirements such as: Personally Identifiable Information (PII), Health Insurance Portability and Accountability Act (HIPAA), Payment Card Industry (PCI), etc. In addition, the privacy definitions and the policy and enforcement effectiveness are themselves varied, complex and change depending on where your data resides – state, province, country. For example the European Union’s (EU) data protection directive (#1) is much stricter than the weak USA privacy laws (note - if you plan to market a global CE product you should follow “safe harbor”).

How does one start to protect critical data and associated privacy requirements with many of the privacy environmental rules and variables themselves in flux? Where lacking common, ubiquitous privacy requirements, few (if any) implementation level, definitive privacy specifications exist for developers to build privacy enhancing technologies (PET), including CE. Therefore we collectively need a global privacy framework to design and measure capabilities, where we chose the “Privacy by Design (PbD)” initiative (#2) as an existing international effort to support. We developed a cyber model that enables the PbD seven foundational principles (describer later). The fair information practices (FIPs) (#3) are another set of high-level foundational requirements that are widely referenced and integrated in privacy rules and laws. As are the Organization for Economic Co-operation and Development (OECD) privacy principles (#4), where both need to be accounted for in a Cyber security For PbD (“C4P”) model. Thus C4P will inherently address the key privacy protection and control aspects from the start, making the actual data environment relatively agnostic to the ongoing global privacy environment churn.

Recently the EU’s top court’s decision on privacy rights (against Google) added the right to be forgotten. We advocate that even the notion of PII (with 12 major attributes) or HIPAA’s Protected Health Information (PHI) (with 18 key attributes) are likely not enough quantification for effective data characterization. As there are 100-1000s+ other identification attributes (from what you search, post) that can pinpoint you; thus is all that data and metadata being collected worthy of privacy protection? Privacy is a simple concept but complex endeavor to protect. Where the protection thereof must be provided from an enterprise view, and start from the source, sensors, then devices and CE, through the networks and ISPs, etc. This end-to-end (E2E) connectivity is where shared vulnerabilities are alive and prospering – especially from the lucrative cybercrime aspect. This article focuses on that enterprise privacy view, where anything that attaches to the “net” can have privacy built in – by design.

Privacy by Design Principles

Given the varying global privacy requirements, we developed our C4P model around the seven major principles in the existing international PbD initiative (Creator, Dr. Ann Cavoukian), also mapping these seven principles to the 26 privacy controls in National Institute of Standards and Technology (NIST) 800-53a Appendix J (#5). Thus our C4P model will inherently address the major privacy protection and control aspects in PhD, FIPs and OCED. Essentially encapsulating the key data protection and control attributes and making them relatively agnostic to the ongoing vague privacy definitions and requirements churn. Current PET and CE products are generally device centric and not integrated as part of an overall enterprise systems of systems (SoS) architecture foundation. Hence current privacy products and services cannot easily integrate into multiple environments or scale – in a continuum from one end device to another, likely different, end device.

The overall C4P technical approach and proposed specifications are described in the “C4P and OPF discussion” section. The seven PbD principles are listed below with general cyber perspectives that are defined in detail in the C4P and OPF section:
1. **Proactive not Reactive; Preventative not Remedial**
   C4P—a cyber model that provides overall implementation guidance, specific methods – enabling privacy.

2. **Privacy as the Default Setting**
   C4P—Many IA / CND products now come preset with secure settings. Management must put security over convenience in configuring the cyber baseline and data protection levels (and still not encumber the users).

3. **Privacy Embedded into Design**
   C4P—Design tenet is build security in, not bolt it on – using common standards, controls, and approved products.

4. **Full Functionality – Positive-Sum, not Zero-Sum**
   C4P—a well-designed cyber model meets these tenets, keeps it simple, does not affect users, and saves resources.

5. **End-to-End Security – Full Lifecycle Protection**
   C4P—IA/Security products are well integrated enablers of applications and data – supporting the E2E lifecycle.

6. **Visibility and Transparency – Keep it Open**
   C4P—Security by obscurity has limited utility, better to provide common methods and products - trust but verify.

7. **Respect for User Privacy – Keep it User-Centric**
   C4P—Security should be an enabler: educate users, manage expectations, and provide for easy state changes.

## Cyber For PbD (C4P) Model Design Rationale

The essence of our C4P approach is to develop an open privacy framework (OPF) using a services-based approach (similar to the platform as a service (PaaS) cloud construct) applying data-centric-security (DCS) methods which are integrated into a SoS package using existing commercial “off-the-shelf” technology (COTS). Our OPF foundation leverages, aligns and is integrated with NIST’s Risk Management Framework (RMF) and Cybersecurity Framework (CSF). By developing and documenting a common OPF that is easy for PETs and CE devices to develop capabilities, C4P enables more integrated privacy capabilities to become available to enhance usability, reuse, and innovation.

The key elements of our C4P model are listed below, along with qualifying aspects and boundaries.

1. **1 - Our hypothesis is that by endorsing and supporting PbD we facilitate a structured discussion on effectively enhancing privacy protection overall, regardless of the environment.**

2. **2 - Data itself can be best protected by an approach covering the application layer to data store schema, where the DCS model most effectively describes that methodology. The related PaaS cloud model also works at the application layer, where C4P encapsulates the controls and data therein (similar to object oriented programming methods).**

3. **3 - An implementation centric C4P model would use a privacy community acknowledged notional requirements set within a universal “use case” that is representative of the privacy ecosphere. We focus on the NIST 800-53RevA Appendix “J” as an initial privacy requirements set. We also leverage the OASIS PbD Documentation for Software Engineers (#6) and PbD related PMRM specification (#7) (including their use case template) to guide our C4P model.**

4. **4 – The C4P approach is built on top of a standard network environment with the typical information assurance (IA) / computer network defense (CND) / security suite. While the C4P protections use data-centric (DCS) methods, the infrastructure still needs its own security to ensure availability and overall system protection.**

5. **The C4P principle capabilities added (+ capabilities) shown in in Figure 1 are:**
   (a) Data security (**DataSec**) (with key management and access control capabilities),
   (b) Software / applications security (**SW/AppsSec**), and
   (c) Security policy architecture and design (**PolicySec**).

   We suggest that any cyber model supporting PbD must also integrate a security continuous monitoring (SCM) / security information and event management (SIEM) capability (**SIEMSec**) to monitor the infrastructure security posture and feedback to capabilities a, b, and c. For existing products that meet or exceed the C4P specifications defined herein, see the various company briefs at: [http://www.sciap.org/blog1/?page_id=1554](http://www.sciap.org/blog1/?page_id=1554).

   The **overall notional C4P approach is depicted in Figure 1**, with capability details and technical specifications described later. A draft, high-level overall cyber model for PbD brief with a much broader IT / risk view is at: [http://www.sciap.org/blog1/wp-content/uploads/Privacy-by-Design-cyber-security.pdf](http://www.sciap.org/blog1/wp-content/uploads/Privacy-by-Design-cyber-security.pdf)
Cyber (Data Centric Security) For Privacy by Design (PbD) = C4P

+ Data Encryption end2end – focused on services / applications (re: PaaS model)
+ Enterprise access control - Multi-factor authentication (re: RAdAC objective)
+ Security Policy management – Automated, serve multiple ‘avatar’ levels in PbD
+ Application engineering - Common model for services, apps, APIs, phones, etc.

Capabilities added on top of the standard IA/CND/Security cyber suite

Monitoring, tracking, assessment = SCM / SIEM, DLP / RBS, Predictive Analytics, etc

Standard IA / CND suite (IA devices) = Firewall, A/V, IDS/IPS, Crypto / Key Mgmt, HBSS, & VPN

Typical Network infrastructure = LAN / WAN (IA enabled devices) = common computing environment

Figure 1: Building privacy protection into the enterprise – from the bottom up.

Implementing Privacy by Design

Any C4P approach to protecting enterprise privacy must do that in a common, fully integrated, easily executable, global manner. IT, Cyber and Data must all be collectively designed, built and operated from an E2E, SoS, fully harmonized approach. There is a natural hierarchy in our enterprise IT/network environment, where the major integration complexities generally arise in the numerous interfaces, protocols and many communications paths typically involved in E2E transactions within a SoS environment. Where the essence of privacy (data protection and access control) is in assuring the information exchange requirements (IERs) between layers / enclaves, and specifically the protections, controls and inheritance aspects therein.

Data Centric Architecture & Security

Central to the overall execution efficiency and implementation consistency of our C4P approach is the general data-centric architecture (DCA) and supporting DCS approach to cyber. These methods are described in several illustrations and descriptions that follow, providing the technical perspective of our C4P model. Data-centric design recognizes that the essential privacy environment invariant are the protection and control of key data and IERs between systems or components. DCA describes the exchange in terms of a “data model” and data producers and consumers of the data. Where DCA / DCS relies on four basic principles: (a) Expose the data and metadata, (b) Hide the behavior, (c) Delegate data-handling to a data bus, and (d) Explicitly define data-handling contracts.

In any C4P model, since privacy is all about control of assured data and IERs, we need to consistently account for how the data moves throughout the enterprise and what the privacy protections & controls are at each layer. DCA decouples designs and simplifies communications and can link individual capabilities in a SoS environment into a coherent whole, using open standards - for one example, OMG DDS (Object Management Group – Data-Distribution Service)(#9). Where details of the transports, operating systems, and other infrastructure information are then not essential to effectively implement DCS. Hence C4P then allows easier adaptation to performance, scalability, and fault-tolerance requirements. Figure 2 depicts a notional E2E DCA/DCS environment.
DCA & DCS Interrelationships

Within the DCA / DCS construct, we collectively need to quantify and modularize the key DCA components and capture the key security specifications (e.g., services, capabilities and profiles). These include (but are not limited to): DCPS, DDSI, DataReader, DataWriter, Pub / Sub. Java, mobile code, widgets, storage functions, middleware, services, ESB, etc., and especially APIs (application programming interface). API numbers will explode supporting IoT with many billions of sensors and devices connected throughout the enterprise, including CE. The overall DCA / DCS interrelationships are depicted in the notional DCA/DCS model in Figure 3.
We propose that an optimum view to depict a C4P model is DCS added on top of the existing IT/network and IA/CND/Security suite used in the typical enterprise environment (shown in the box above). As initially described, we view this C4P approach best as a services-centric, PaaS-like model, where the data, applications and controls are encapsulated in an essentially “by session or transaction VPN” approach and thus agnostic to the overall infrastructure vulnerabilities. We account for the E2E access control and security policy details to satisfy the privacy elements later in our “C4P and OPF discussion” section. There are many benefits of a DCS approach within the PaaS model (#10) that make it a very useful design methodology for any C4P approach.

Figure 4 depicts the various services-based models that we use to describe our C4P. The key point in the using the PaaS representation for C4P is that by protecting the applications and data layers, sensitive information is then inoculated from many, if not most, of the vulnerabilities in the lower layers. The overall systems availability still needs to be preserved by the infrastructure layers, thus it’s still essential that the typical IA/Security/Cyber suite be effective and maintained (and well monitored using SCM / SIEM).

Our cyber community’s collective privacy (data security) architecture goal needs to include a common Trusted Cyber Infrastructure (TCI) within “an adequately assured, affordable, net-centric environment” built from disparate heterogeneous capabilities that we integrate into a homogenous cyber ecosphere. The cyber end-state would be to drive IA / CND / security capabilities into a true commodity state, by using IA building blocks, approved products, APLs/PPLs (e.g., existing “NIAP” programs). Where interoperability and composeability are built in upfront (using DCA) and help dramatically reduce complexity and ambiguity. Thus establishing known risks and pedigrees which reduces the attack surface, risks & TOC; where this TCI model with EAL capabilities built in becomes the security infrastructure baseline for PbD. Our C4P approach also subscribes to the NIST ‘building in security” methods for a TCI (e.g., SP 800-160, Systems Security Engineering: An Integrated Approach to Building Trustworthy Resilient Systems [http://csrc.nist.gov/publications/drafts/800-160/sp800_160_draft.pdf]). A notional TCI environment illustration is shown in Figure 5:

![Figure 4: Various Cloud views – highlighting the PaaS approach in C4P](image-url)
What “really” matters in Cyber

What “really matters” then in implementing a common, affordable E2E TCI across all organizations? While the easy answer is “it depends on the environment” - there are several key cyber factors we all collectively still need to build in to our environments and better support PbD too. It’s generally accepted that we have reasonably effective IA/Cyber technologies available now (to at least a first and second order effect). Where we all just need to integrate and maintain the cyber suite a lot better (principally using enforced cyber hygiene and effective access control).

We base our C4P model added capabilities and approach on the factors described so far – also subscribing to the four essential cyber capabilities that will endure as called out in the “Enterprise SW / SOA IA / security approach (clarifying the fog of IA)” paper (#11). The four main thrusts in the executive summary, page two, are still germane now: thus we embed these four technical aspects into our C4P approach:

1. Use IA/cyber standards and related profiles within a functional cyber architecture,
2. Provide E2E enterprise access control (IA&A) using an implementation centric approach (e.g., ZBAC),
3. Use a data-centric security approach (along with potentially adding content based encryption), and
4. Provide dynamic security policy execution amongst operations, management, etc.

C4P and OPF

Each of the “+ capabilities” functional details (re: listed on page two figure 1 and tenet number four: DataSec, SW/AppsSec, PolicySec and SIEMSec) are provided in depth. This quantifies the support for the privacy capabilities stated in “Operationalizing PbD (O-PbD)” document (#12) – specifically supporting the seven key principles listed earlier. A detailed explanation of C4P is necessary to provide a more complete description on how each capability fits each principle. We start the C4P story by using the O-PbD privacy definitions: “Privacy is about … maintaining personal control over the collection, use, and disclosure of one’s personally identifiable information” (“informational self-determination”). Privacy is indeed a complex topic by itself, let alone the USA vs EU differences. The O-PbD privacy perspective discusses many process related aspects of PbD as well: (a) compliance, (b) process improvement, and (c) privacy policies need to be baked into applications across the entire software development lifecycle (SDLC). As stated earlier, the automation and embedding of security policy to manage the
controls needed in various privacy levels and environments is the most critical aspect of any cyber model for PbD. We all must of course build any cyber model to the same privacy requirements, ideally quantified in privacy specifications derived from common, approved, authoritative privacy sources.

**DataSec**

*DataSec (end-to-end encryption, data-centric services, key management and access control)* capabilities for PbD:

**User security**: PbD requires that only authenticated and authorized users have access to the privileged parts of their PbD enabled applications. To restrict access to other users, DataSec provides multi-factor authentication which cover location, time, biometrics and other sensor data from the user before allowing access to the more sensitive parts of a PbD enabled application.

**Security against data breaches**: Data breaches are now a routine occurrence and with the proliferation of cloud computing much of the sensitive data processed by PbD aware and enabled applications is now resident in cloud datacenters. A PbD application must have data-in-transit and data-at-rest security at the back end but also allows the trust footprint to be smaller. This means that the database servers, file servers, the administrators, data center technicians or any intermediate equipment can all be untrusted.

**Better operational awareness**: Security exceptions for both user security and data security are logged for audits and outlier events raise alerts to users and application owners.

**SW/AppSec**

*SW/AppSec (Apps / Services & Phone/Mobile)* capabilities for PbD:

**Automated**: PbD dictates for the need of automated policy authoring, enforcement and auditing. If the security is based on manual processes, then points of error, vulnerabilities, and noncompliance are likely to be created.

**Ubiquitous**: Omn pres ence - the same control and management implementation should be operable on any environment, regardless of physical location, operating system, virtualization platform, or deployment method used. Policies defined by PbD should accommodate all the entities (hardware and software) and their operation (message exchange, file storage, etc.) within the environment.

**Scalable**: The system should automatically grow and contract to meet the changing demands of applications and underlying infrastructure. PbD must automatically provide the appropriate security controls and maintain appropriate threat and compliance monitoring as infrastructure environments scale up or down.

**Multi-Layer Visibility**: Privacy challenges exist in both hardware and software. Any SW/AppSec solution considers privacy as an integral part of security and hence provides comprehensive solutions at each operational hardware and software layer.

**PolicySec and SIEMSec**

*PolicySec and SIEMSec (Security policy architecture and SCM / SIEM)* capabilities for PbD:

**Policy authoring**: PbD needs an intuitive, user-centric privacy policy authoring feature for users to set their privacy policies (“informational self-determination”). Where PolicySec must provide more functions than just enforcement of access control policies.

**Policy enforcement**: PbD needs a tool that maps these intuitive privacy policies into technical enforcement (access control, confidentiality etc.) across the information lifecycle and software development lifecycle, and configurable “privacy code libraries.” Attribute-Based Access Control (ABAC) and encryption are example mechanisms that can be configured to enforce the privacy policies.

**Policy audit**: PbD needs a user-centric tool that lets users verify (audit) that their policies are enforced correctly. PolicySec & SIEMSec help audit “as-is” processes & controls against the defined security policies for privacy.

**Model-Driven Security Policy Automation** (“MDS”) is the tool supported process of modelling security requirements at a high level of abstraction, and using other information sources available about the system (produced by other stakeholders). These inputs, which are expressed in Domain Specific Languages (DSL), are then transformed into enforceable security rules with as little human intervention as possible. MDS explicitly also includes the run-time security management (e.g. entitlements/authorizations), i.e. run-time enforcement of the policy on the protected IT systems, dynamic policy updates and the monitoring of policy violations.

**Model-Driven Security Accreditation Automation** (“MDSA”) automates the analysis of traceable correspondence between technical security policy implementation (e.g. ABAC) and the information assurance requirements captured in “undistorted” requirements models (e.g. Common Criteria, control objectives). MDSA also
documents “supporting evidence” for accreditation based on various information (esp. design-time system / security models, system / security artifacts, system / security model transformations, and runtime system / security incident logs). Furthermore, MDSA enables the automated change detection and analysis to determine whether the accreditation is still valid.

**Information Lifecycle View**

The essence of our C4P approach is to develop an OPF using a service-based, PaaS-like, approach applying DCS methods which are integrated into a SoS EA foundation using existing COTS products. The following C4P high-level reference implementation management approach, shown in Figure 6, includes functional design specifications for each “OPF-xx” capability shown. Each function includes a “traceability” aspect to assure compliance with all requirements through product lifecycle – as well as detailed specifications for each that allows developer’s more definitive build guidance.

Privacy protection can be viewed as an information lifecycle governance/management problem. Privacy policies need to be enforced for information, including collection/creation, access (incl. delegation), transmission, storage, redaction, deletion, expiration etc. In addition to numerous non-technical controls (discussed in the O-PbD literature), a number of technical information security features need to be implemented. Our reference implementation technical approach is built on unique combination of a set of common technical components, for which reference implementations are already developed and deployed by the partners. Our integrated OPF EA reduces the “fog of privacy requirements” (e.g., being infrastructure agnostic); thus simplifying the overall privacy ecosphere and facilitating developers, PETs and CE companies develop interoperable privacy capabilities.

![Figure 6: Reference Implementation Combined Architecture Picture](image)

The main of technical components and functions in the information lifecycle reference implementation are:

**OPF-PM: - Policy Management** - PbD needs a manageable intuitive, user-centric privacy policy authoring feature for users to set their privacy policies (“informational self-determination”) governing users, systems, applications, and interactions (information flows). It needs to allow users and administrators to author and/or select privacy policies captured in intuitive models (OMG-style Domain-Specific Languages, DSLs). MDS takes the privacy model, the generated system description, and other information as inputs into the MDS “model
transformations” and automatically generates configurations for the various other components of the solution, and fine-grained access rules (which are information-flow based and attribute-based). To solve the management challenges of attribute-based access control (ABAC, described below), and to turn human-intuitive, generic PbD policies into technically enforceable policy rules, we recommend the use of “model-driven security” (MDS) policy automation approaches: MDS helps simplify and automate security policy authoring and management, and automatically generates/updates fine-grained technical policy rules for the full technology stack. MDS is the tool supported process of modeling security requirements at a high level of abstraction, and using other information sources available about the system (produced by other stakeholders).

**OPF-PE: Automated Security Policy Enforcement & Alerting** - PbD needs a tool that enforces technical privacy rules and configurations generated by OPF-PM technically (access control, confidentiality etc.) across the IT landscape (multiple layers of the system/application/network/VM etc.), across the information lifecycle and software development lifecycle. Attribute-Based Access Control (ABAC - which evaluates attributes associated with the subject, object, requested operations, and, in some cases, environmental conditions against policy, rules, or relationships that describe the allowable operations for a given set of attributes) and encryption are example mechanisms that can be configured to enforce the privacy policies. A good proportion of privacy policies can be technically implemented using fine-grained, contextual access control rules. ABAC rules are enforced using PbD appliances and/or locally installed Policy Enforcement Point (PEP) software proxies. MDS solves one of the main challenges around ABAC’s various management and implementation challenges. Note that the solution does not just grant/deny information flows, but also has to support redaction & filtering - which is challenging, because the PEP needs to be highly aware of the specific data being requested and provided, so it can apply the redaction and filtering. Furthermore, the Policy Decision Points (PDPs) need to support the policy features that can capture generic, but detailed enough redaction and filtering instructions. As the diagram illustrates, we implement a redaction & filtering service alongside the PEPs, which query the data source and instruct the PDP/PEPs. This architecture is designed for flexibility. Another aspect of this feature is to alert the proper people that something is happening that needs attention. It can also be set up to take action such as blocking traffic to or from a malware or adversary. The solution’s security rules are customizable and reactive to allow tuning to your network environment.

The following diagram illustrates how the policy enforcement points (PEPs) implement redaction and filtering capabilities. The main challenge is that the PEP needs to be highly aware of the specific data being requested and provided, so it can apply the redaction and filtering. Furthermore, the Policy Decision Points (PDPs) need to support the policy features that can capture generic, but detailed enough redaction and filtering instructions. As the diagram illustrates, we implement a redaction & filtering service alongside the PEPs, which query the data source and instruct the PDP/PEPs, designed for simplicity.

**OPF-CM: Compliance Management & Automation** - PbD needs a user-centric tool that lets users verify (audit) that their policies are enforced correctly. This feature analyzes the traceable correspondence between technical security policy implementation (e.g. ABAC) and the information assurance requirements captured in “undistorted” requirements models (e.g. Common Criteria, control objectives). It also documents “supporting evidence” for accreditation/compliance purposes. It helps audit “as-is” processes & controls against the defined
security policies for privacy. It uses “model-driven security” accreditation automation approaches to automatically correlate, analyze and document the traceable correspondence between technical security policy implementation (e.g. ABAC, information flows, incidents, assets, security policies, and other information.) and information assurance requirements captured in “undistorted” requirements models (e.g. Common Criteria control objectives).

**OPF-SD: System (of Systems) Discovery** - The system automatically generates a model of the enterprise networks, systems, applications, information flows, users etc. This “system description” plays a similar role as Common Criteria’s “Target of Evaluation”. Note that a system description could also be manually modeled for very structured, critical IT landscapes. However, in most environments, automatic detection will be the preferred choice because it is much easier/cheaper. The system needs to support this capability to discover assets so it knows what to protect (computers, routers, switches, etc.) and find network’s entry and exit points. Its enterprise network mapping (ENM) feature captures network knowledge at the data link and network layers using passive monitoring, OS fingerprinting, SNMP, and customizable active probes to populate the asset database. This is used to monitor asset state and bandwidth usage.

**OPF-IM - Incident Monitoring**: The solution needs to be able to watch network activity (including bandwidth usage), access control incidents, and more, by capturing automatically captures and analyzes anomalies detected in PbD appliances and/or locally installed Policy Enforcement Point (PEP) software proxies. As an intrusion detection system (IDS) and intrusion prevention system (IPS), the solution performs signature matching, anomaly detection, blocking, and other analysis in real-time. Its Leak Detection\Defense (LDS) System is able to stop ex-filtration via several customizable lists. Its anomalous bandwidth analysis identifies unusual spikes in bandwidth usage.

**OPF-PS - Presentation of (Current) Status**: - The solution displays the current privacy posture on a continuous basis in a consolidated fashion. This includes the network status (e.g. in a web browser or its 3D asset viewer), a dashboard that reports on levels of events with options to drill into details - even the triggering network packet, a policy incident viewer, a compliance evidence viewer etc. These events need to be categorized and graphed to display the state of the SoS.

**OPF-SC - Security Administrator Collaboration**: The solution also includes a way for administrators to collaborate to resolve issues (e.g. a secure social network to facilitate collaboration between administrators.

**OPF-ER - Encryption for Data at Rest**: The solution also needs to protect information at rest using encryption. The cryptography is configured and managed in a unified way together with the other policies in OPF-PM. The cryptography should be at least NSA Suite B certified and should not only encrypt the data for privacy but should also have checks for data integrity. Data should be classified into various isolated, application layer defined security domains (e.g: research group, operations group, common etc) and every effort should be made to divide security domains into smaller, more specific security domains (e.g. research-unit1) to practice the principle of least privilege or “need to know” basis. Security domains never share data encryption keys and themselves have access-control-lists which restrict who gets what type of access to data within a security partition. Every data element in the entire system belongs to exactly one security partition which shared data resident in a security partition dedicated to facilitate such sharing across groups. Any access violations or integrity check failures are immediately logged for audits and machine intelligence generates the appropriate prioritized administrative alerts. Finally, the data encryption keys should not be resident with the data being protected and the data encryption keys should themselves be encrypted for privacy and integrity by elliptic curve NSA Suite B certified cryptography.

**OPF-ET - Encryption for Data in Transit**: The solution also needs to protect information in transit using encryption. The encryption is configured and managed in a unified way together with the other policies in OPF-PM. Data in transit between storage and processing or between processing elements may be protected by SSL for an outer layer of encryption but must have an inner layer of encryption to be protected similar to the provisions in OPF-ER. It must therefore preserve the OPF-ER requirements in terms of NSA Suite B cryptography, privacy as well as integrity checks, security partitions, access-control lists, audits, prioritized alerts. Data in transit between processing and user presentation may continue to rely on SSL while underlying ciphers in use are NSA Suite B compatible.

**OPF-AH: User/Machine Authentication**: The solution needs to also support the appropriate level of authentication. User Authentication should be based on 5 factors, namely the user memorized password or PIN, a cryptographically secure time-based one time password or token, successfully matched facial patterns of the user, location of user as well as time of request by user. The system may be dynamic to allow for fewer factors to be
imposed on the user when the risk is determined to be low (e.g. access from within a secure network operation center) while maintaining all five factors (or more) when the risk is determined to be elevated. All factors must be bound to a single authentication session attempt and must be protected against replay attacks.

Summary

Privacy matters everywhere, IoT and CE included, where the downside of not protecting privacy is greatly increased individual, company and organizational risks which are costly and last forever in the public domain. While we focused on the enterprise privacy protection methods herein with our C4P approach, the basics elements of trust and data protection must be applied at the end devices as well, supporting E2E privacy. The threat vectors are too numerous to chase and try to fix / patch individually (and they morph and change by the hour); thus privacy must be built in by design, from the start, from the end-points through the ISP. It does little good to fix the SCRM aspects mentioned at the beginning for CE devices, when what you connect to is vulnerable – as privacy is still lost. In the legal world we live in, cyber third party suits will become the norm, especially with data breaches; thus a poorly secured CE device is a huge financial risk (just as you must follow the UL rules, so will you need to do for privacy).

Our proposed C4P model and accompanying OPF EA is an executable foundation to build interoperable secure privacy capabilities into any standard IT Network environment with the typical IA / Cyber suite. This enterprise view starts from the sensor data methods, through devices / CE, the network protections and connections to the internet. The OPF EA can be used as a vehicle to help update the privacy specifications that developers need to build privacy capabilities into a collective cyber ecosphere, providing guidance for PETs and CE. C4P provides an E2E, SoS, OPF that can scale, adapt and endure and work well in most environments. In addition, this C4P model can be implemented now, while the global privacy requirements and of the rest technical world catches up to making privacy a priority. Thus minimizing long term privacy liabilities and costs in the interim for users and companies.

References and Bibliography:
1 - Data Protection Directive 95/46/EC  
2 - PbD - Privacy by Design – seven principles  
http://www.privacybydesign.ca/index.php/about-pbd/7-foundational-principles/
3 - Fair Information Principles (FIPs) Practices  
4 - Organization for Economic Co-operation and Development (OECD). Guidelines on the Protection of Privacy and Transborder Flows of Personal Data  
http://www.oecd.org/sti/ieconomy/privacy.htm
5 - NIST 800-53Rev4 – Appendix “J” – Privacy controls  
http://dx.doi.org/10.6028/NIST.SP.800-53r4
6 - OASIS Privacy by Design Documentation for Software Engineers (PbD for SW SE) (DRAFT!)  
http://docs.oasis-open.org/pbd-se/pbd-se/v1.0/cs01/pbd-se-v1.0-cs01.doc
7 - OASIS - Privacy Management Reference Model and Methodology (PMRM)  
http://docs.oasis-open.org/pmrm/PMRM/v1.0/cs01/PMRM-v1.0-cs01.doc
8 – National Information Assurance partnership (NIAP) – Product Compliant List  
https://www.niap-ccews.org/CCEVS_Products/pcl.cfm
9 - Object Management Group – Data-Distribution Service (OMG DDS)  
http://portals.omg.org/dds/
10 – Platform as a Service (PaaS) cloud model  
11 - Enterprise SW / SOA IA / security approach (clarifying the fog of IA)  
12 - Operationalizing PbD (O-PbD) –  