TITLE: SECURE RESOURCES WITHIN SMART ENVIRONMENTS

Sailesh Sathish, Jilles Van-Gurp, Christian Prehofer
{firstname.lastname@nokia.com}

1. INTRODUCTION

The emergence of pervasive spaces, or smart spaces, and the ongoing integration of those with each other over the internet is creating an internet of things where people interact with each other, the things connected to the network (locally or via internet), with applications and services provided in the internet, or all of these at the same time. The security needs of the internet of things are well known. However, the state of the art of of today's internet technology only addresses these to a very limited extent. Current web technology will need to evolve such that more complex interactions in the internet of things can take place securely. Browsing the Internet on a standalone PC has now become "traditional". As the web is becoming more pervasive, the push to new platforms such as mobiles has happened, driven in no small measure by popular consumer demand. Coupling mobility to traditional browser paradigms, this is pushing the web into the realm called the "mobile web". While arguments go on about whether there is a real "mobile web" or giving the users the "whole web" experience on mobiles, mobility does bring new aspects that enables novel adaptive applications for the web. Coupling mobility to applications brings about pervasive outreach where applications can harness the changing environment of the user. The situational characteristics of the user and device can be combined to enable new genre of adaptive applications that have so far not been possible through "traditional" PC oriented systems. In order to enable such outreach, device manufacturers are coming out with API functionalities that enable applications especially web applications access to user and device properties. An example is the Nokia API for Series 60 widgets. On another front, devices now come with a suite of proximal networking technologies. Examples are Bluetooth, Wibree and WLan. The advent of smart spaces provides new sets of dynamic properties and services that can be leveraged by any adaptive applications. Devices can access these through proximal networks exposing them to local applications (or directly to user) via the runtime. This goes way beyond the current offerings of a fixed device API set. Keeping in mind our current and near future capabilities, the issue of security and privacy is at the forefront of all bottlenecks.

We pen some thoughts on privacy and security issues from two perspectives while viewing them as interrelated. The first is the internet of things that looks at authentication and authorization of users and web resources. The second is dynamic interaction on local device and smart environments. We present a few challenges that has to be taken into account while formulating any framework for secure access to device API's. The intent is to offer those scenarios that need to be satisfied in order to ensure efficient models that are progressive enough. The security framework thus need to be extensible enough so as to ensure compatibility across a wide range of applications from simple data access to more complex bi-directional communication with resources that work within smart environments.

2. INTERNET OF THINGS

We see the following challenges regarding extending current internet technology for use in the internet of things:

- Authentication. Proper authentication of all parties involved in a particular interaction is the base line for any sensible security system. Authentication in the internet is currently focused on authenticating users using a browser. Authentication needs to be extended to also cover things, applications and services. This is important because in order to interact securely, all parties involved with the interaction need to be able to establish who they are interacting with in order to be able to authorize access.
- Federated identity. A problem in the current internet is that few sites collaborate on identity. This means that users need to create and maintain user profiles with each site they interact with. This is problematic today already and the escalation of this problem with device and vendor specific means of authenticating in the internet of things will create an unacceptable end user experience that will complicate user adoption of otherwise compelling use cases.
- Many to many authentication. Current technology is focused on security between two interacting parties. However, in the internet of things, there are many people, devices and services involved that engage in highly complex interactions with each other. Consequently there is a need to do so securely and in a non intrusive way. While technically, 1 to 1 type interactions can be combined to perform n to m type authentication, doing so with current technology would result in excessive and repetitive user interaction or in excessive amounts of network trafic.

- Reusable policy. In the internet of things there is a need to apply policies across services, resources and devices controlled or owned by a person or institution. Due to the amount of things in the internet of things, doing so on a case by case basis like it is common today, quickly becomes unworkable. Consequently there is a need for being able to create and reuse sensible high level policies that govern e.g. access to a user's private data and context, access to devices in a particular building or room owned by a company, etc.
- Session management. In the internet of things, people, services, and devices may be engaged in multiple interactions at the same time. Each of these sessions has a security state associated and the flow of information between sessions needs to be carefully regulated. A simple example in today's internet is the need to shield Javascript access to browser state of different browser windows. While most browsers do a decent job of this, browser security bugs can break the intended behavior and expose users to phishing attacks and other scams.

We believe these challenges are strongly related to each other and need to be addressed in a coherent, scalable, backward, and forward compatible way to allow the existing internet to evolve such that current applications and services in use today can start integrating with each other and with things connected to the network.

3. CONTEXT AND SMART ENVIRONMENTS

Recent market entries especially the smart phones come with more computing power, memory and a wide array of proximal and remote networking capabilities. Certain categories also come with integrated sensors such as accelerometers, temperature, orientation etc along with new interaction capabilities such as touch screens, gesture and voice user interfaces. So, what do we do with all such capabilities? Platforms are already available or are being rolled out to market that enables access to such properties. Applications are able to access them individually or in combination to get to what is known as "Context". Context defines the state of the user at any point of time. Platform middleware can support building combinations or abstractions of one or more "raw" or "native" data streams and performing reasoning to describe current user state. Examples of such are presence information like "I am at office", "I am driving" to simple device states such as "keypad locked" or "camera turned off".

3.1 Smart Spaces

A smart space is a multi-user multi-device dynamic interaction environment that is aware of its physical environment working on top of heterogeneous radio technologies/software distribution platforms. Several devices and services can interact within a smart space environment providing new interaction services and adaptive applications to users. Smart spaces are highly dynamic environments where new services can be discovered and released in accordance with user movements or other dynamic criterion. Smart spaces range from having dedicated infrastructure and services to ad-hoc spaces formed through proximal interaction of smart devices. Whatever be the model of service provision, smart spaces are characterized by dynamic provision of services and resources that can be directly provided to users or utilized by services running locally on participating devices. Providing support for smart applications include supporting a smart API model that is capable of supporting dynamic sessions including addition and removal of new services from the model. Operating within a dynamic environment brings up additional requirement on security including group authentication and authorization that will have direct impact on the way policies are implemented within the environment.

We use the term "Providers" to describe the sources of context information. Providers can provide either a single raw data such as a GPS coordinate or abstractions such as presence. The term "Consumers" are used to refer to applications that use context data to perform some service. Providers can reside (at least in theory) anywhere. Providers can be part of your local device or it can be accessible through a network. Providers can be a set of static properties that are always available or they can be dynamic so that they are available only when the user is present at a particular place. The notion of "smart spaces" promotes the idea of dynamic providers where services can be discovered and accessed within spatial dimensions. Providers can expose data to consumers where the data itself is static or they can expose dynamic data. It is thereby important to understand the difference between "dynamic providers" and "dynamic data providers".

3.2 Challenges in dynamic environments

The amount and type of security and privacy is also dependent on the capability and design of the API itself. Browsers typically employ a binary mode of secure access for information share between different resources. These are either full access as in the case of inline scripts from other applications or no share at all in the case of applications running within multiple frames. For web access to device API's a binary model is not the preferred way of working. Within dynamic environments, we are exposed to set of platforms that can provide:

- Different modes of access to properties – object based approaches vs data based approach
- Providers that can provide static and/or dynamic data
- Providers who are themselves dynamic i.e. supporting dynamic sessions
- Providers that are locally resident or remotely available

With such platforms, the following scenarios are possible:

- Local properties providing data to Remote devices
- Local properties providing data to Local device
- Local properties and Local applications
- Local properties providing data to Remote applications
- Remote properties providing data to Remote applications
- Remote properties providing data to Remote devices
- Applications to Applications – Local and Remote

Another issue to be considered is the API design and the way properties accessed by applications especially browser applications. Will it be a single instance of a particular property (provider) that will be shared by browser applications (consumers) running within multiple tabs or frames? Or will each web page get its own instance? Again, the issue of single or multiple instances will depend on the functionality provided by the API. If for example, only a read functionality is provided as in the case of for example, GPS data, a single instance may suffice. If read, write and executable is needed and these can be serviced only by individual instances, then multiple instances of the same property is needed. Of course it is possible to service multiple applications requiring customized services through a single instance but there are exceptions. In the case of a single instance with READ only functionality, consumers will read the same value from providers. In providing multi-application support, one major problem is when interpretation of a value under a specific application context might become different between applications. For example, a UI widget might provide data meant for increasing the volume of a media player while a list selection application might interpret the same UI state as different. This would lead to undesirable executions. One method to solve this problem is the notion of temporary ownership similar to thread locks in software engineering paradigm. This would lead to notion of having a 1) Original owner of the property and 2) Temporary owner of the property

3.3 Profile based approach

In the case of Multidevice smart spaces, where resources are shared between devices, we identify three high-level roles that come into play in the case of providers. We assume a profile based approach where profiles dictate the authorization for applications to each property that can be accessed. The profile is constructed based on a set of categories to which each provider can belong. This is different from the general way of thinking where only consumers are provided with sets of rights. In smart spaces both consumers and providers form dynamic sessions and the middleware acts as a brokering platform. The middleware provides managed pipes that perform rights management for both providers and consumers. Like providers, consumers also belong to categories that define what their specific rights are. For the simplest scenario, where the available API set is completely static as is now the case, this is usually provided by the device vendor at install time with full rights for providers. This must not hold true for smart environments. Similarly consumers can also belong to category sets that may be different to provider categories. Continuing from previous section, we have the roles of original owner of property

and temporary owner of property. For clarity, these are called *property owner* for owner of property and *information owner* for temporary owner for property.
The property owner can set the following:

- Whether this property can be shared to local or remote devices
- Whether the information can only be owned by certain categories of applications
- Whether the property can be shared via multiple or a single hop (i.e property can be forwarded to another device)
- Level or accuracy of information that can be passed to receiving entities
- Grant write access to information owners

The information owner can provide an additional URI or inline policy by accessing the property instance within the middleware. The information owner cannot override the policy of the property owner. The information owner can set the following:

- Whether current information is shareable to all local, remote applications or both
- Whether current information is shareable to particular categories of applications
- Whether current information is shareable to specific application ID's
- The set of applications or categories to which this information holds true

The information owner can only set these values to the property (provided they have write access) and it is upto the middleware to decide how the behavior should be. For example, a device middleware can simply list this information as meta-information to the property while another can read these rights and totally mask the property from those property to whom read access cannot be granted. This sharing is only subject to whether the property owner has set shareable access to local and/or remote devices.

In addition to these roles, we also have the device role that is applicable to all properties that are supported by the middleware. The device role provides the Device Access Policy (DAP). The DAP describes the access policy that devices can provide to the middleware that are applicable for all properties that the middleware hosts. Since the assumption is that one device can host only one middleware, the DAP can be seen as synonymous to middleware access policy. The DAP policy is dynamic and can be updated by the middleware during each session probably with consultation with the user. The behavior of DAP management is up to the middleware implementation. For example, the middleware can maintain history of access patterns of the user (provided authentication has been performed for each user) and cache policies or it can start afresh for each session. The DAP can provide the following

- List of properties that can PUT (WRITE and CREATE) data into device
- List of devices that can PUT properties into device
- List of devices that can connect to middleware
- Whether multi-hopping of properties is allowed
- List of applications that can READ data from the middleware
- Whether direct or multi-hop properties are allowed CREATE access
- Authentication of properties is required or not
- Access to devices based on ownership (use of OpenID/OpenAuth)

The position paper does not refer to consumer policies for information access. The intent is to bring to attention the need to support dynamic environments and the problems that have to be addressed. We acknowledge that the above listed ones do not form the full set that has to be addressed but rather a small subset that should be provided for. Any policies that are formulated must take into consideration support for dynamic properties and provide for extensions to support such advanced scenarios. There must be support within the security framework that would address both consumers and providers within such dynamic environments.