物联网中间件面临的主要挑战及目前主流解决方法

1. **IoT Middleware面临的挑战**

1、The ﬁrst challenge is in developing an IoT middleware that must be available in the cloud as well as on the edge (IoT devices and gateways) for supporting all types of IoT applications, for better privacy control and latency.

意思就是，我们需要开发一个IoT中间件，它必须在云中以及设备和网关上提供，用于支持所有类型的IoT应用，以实现更好的隐私控制和延迟。这就要求系统必须便携且轻量级。这需要中间件开发人员，在高性能服务（例如基于语义的发现、流处理）与在受限设备中部署IoT中间件之间进行权衡。**很显然：在底层设备部署中间件可以控制通信，从而进一步增强隐私安全。**

2、The second challenge is to empower consumers to create IoT applications targeted to their context.

意思就是，中间件需要向使用者提供能够创建面向情景语义的IoT应用程序。Composition Engine (也就是复杂事件处理CEP中的Rule Engine)是必须的，使得用户可以整合云端的语义服务、来自网关的数据服务以及其他的IoT设备等等设计自己的应用语境。

3、The third challenge is to provide semantic service discovery that goes beyond discovery of IP addresses of the nearby IoT devices.

也就是服务发现机制。考虑到物联网环境不断发展，新的服务或设备可能随时上线，同时，现有设备可能随时失效。中间件必须能够在设计时和运行时正确的发现或查询兼容的服务。例如，在一些关键的健康监控IoT应用程序中，必须实时更换发生故障的IoT服务，而不能造成任何中断。

4、The fourth challenge is to guarantee the security of IoT applications and also protect the privacy of users.

老生常谈，隐私和安全问题。

1. **现有IoT Middleware服务发现机制**

A common approach to support adaptability in IoT middleware is to adopt a uniform service description and a mechanism to reason over that description for service discovery in different contexts. For example, at design time, a user should be able to query a registry of service descriptions and with the click of a few buttons start to interact with a device that can guarantee the desired capability and quality.

意思呢就是采用统一的服务描述机制，方便服务发现基于情景进行相应的推断。举得例子就是，用户请求服务描述的注册表，然后通过很少次的点击选择，就可以得到保证期望的能力和质量的设备。**= =还是需要人为参与的，根据人的需求得到相应的服务和设备，可以考虑通过machine learning进行最佳推断。**

Note that: 可用于提供IoT设备的语义注释的开源本体工具（例如Protege）

1、Semantic Sensor Network，SSN 语义传感器网络本体，是由OpenIoT项目提出的。SSN使物联网设备和观察到的数据以可被他人搜索和理解的方式进行注释。在XGSN中，GSN的扩展，物联网设备和观测数据都使用SSN进行语义注释。传感器的语义注释包括与物理传感器相关联的元数据，例如其位置，其产生的观察类型，传感器的负责人或组织以及源类型。其他语义注释与观察数据相关联，包括观察发生时的时间和语义，观察属性，测量单位和值。带注释的数据转换为RDF流，并发送到cloud-enabled Linked Sensor Middleware（LSM），链接数据RDF存储仓库。LSM可以将RDF流数据存储在三元组中，或者使用SPARQL查询对流数据执行连续查询实时导出聚合的观察值。每个GSN虚拟传感器在LSM的RDF存储中都有相应的传感器实例。 XGSN提供RESTful的虚拟传感器的注册服务。

2、SensorML通过提供一种使用标准化XML标签来描述传感器的方式来支持IoT设备的语义描述。其实与上面所说的方法大同小异= =。其主要目的是提供传感器，致动器和处理器的语义注释，使得它们可以互操作，并且可以被机器理解并在复杂工作流中被自动利用。 SensorML is a part of the OGC Sensor Web Enablement (SWE) suite of standards.

3、Hydra采用的是OWL和SAWSDL对IoT设备进行语义注释，使得能够被发现。OWL是针对语义Web的本体语言。OWL本体通常也是存储为RDF。The Semantic Annotation for Web Service Description Language (SAWSDL) allows WSDL (Web Service Description Language) ﬁle to be enriched with semantic description. This assumes that IoT devices can be automatically exposed as WSDL ﬁles.

4、还有一些系统，连接到物理IoT设备的传感器节点按照关键词来携带设备的文本描述。 然后，用户可以通过匹配包含关键字列表的查询来找到这些设备。类似系统有：Snoogle , Microsearch, and MAX

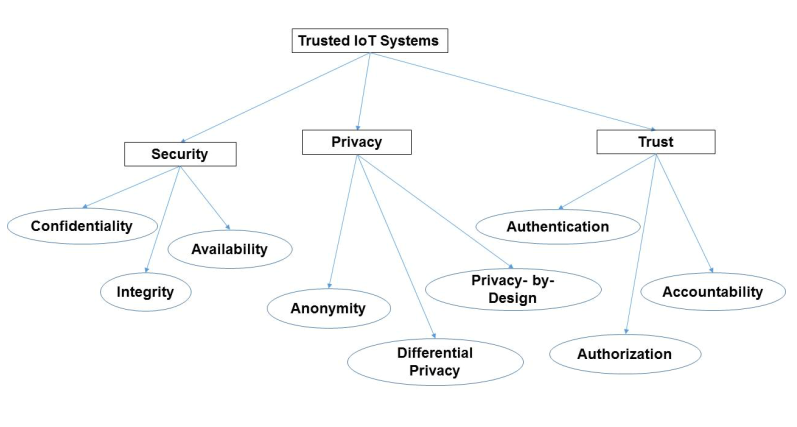
5、Dyser uses statistical models to predict the state of its registered resources when a user submits a query. The predictions are used to establish a ranking that determines the order in which resources are contacted to ﬁnd out whether their current state matches the query. However, contacting registered resources for every query implies serious performance overhead, particularly given that the number of queries and things could be enormous. It is not trivial to build an efﬁcient index for such a big data set. 怎么说呢，Ostermaier的Dyser系统，尼玛就是我设想的那种方法= =，通过统计模型来找出适合用户请求的资源，问题是，为每个查询去contact资源进行相应的判断意味着严重的性能开销，特别是查询和事情的数量可能是巨大的。为这样的大数据集建立有效的索引并不简单。

**上述方法的缺点：当在IoT中使用本体方法进行服务发现时，还存在可扩展性问题。 当存在大量传感器在长时间段内流传输大量数据时，为了获得数据的语义，RDF可能太冗长。 Furthermore, SPARQL query engine used for querying the RDF store might not scale with the large number of data streams from IoT devices. 另外，维护一个描述所有IoT 设备的Global Ontology是不现实的，各个系统所用的ontology不尽相同，设置一个普适的注释工具或者进行系统整合都有很大的困难。**

6、non-ontological服务发现方法之一：为异构设备或者服务，建立一个基于相似度的搜索引擎。using the Earth Movement Distance (EMD) based similarity search algorithm, 对IoT设备的接口文档进行搜索。文档格式不必须一定，但肯定要有一定的规律性。局限： 这种方法被限制用于发现传感器的元数据，而不是来自传感器的观察数据。例如，发现在过去两个小时内读数高于90度的所有温度传感器服务的查询无法使用此方法回答。

7、第二种方法是：基于对设备或服务的使用日志执行数据分析。此方法旨在收集设备和服务的使用历史记录，并使用它来推断设备之间的功能和关系。通过分析何时，哪里，哪些人正在使用该设备，可以导出设备和服务之间的关系。尼玛，有用到了统计= =，再进行智能推荐什么的，感觉就很容易了。

1. **IoT Middleware安全模块**



具体相关的属性如上图。