www.nornir.no

Real Time Web

Unleashing the Full Potential of the Internet of Things

White Paper

Oslo · November 2017

Introduction

As the Internet changed our world profoundly, the Internet of Things is set to change it again, but this time the change will be even deeper and more pervasive. Consumers and businesses will benefit from new types of services and better quality of services, leading to improved convenience, health, and safety.

Many new business opportunities will emerge from this technological change, paving the way to whole new business areas. Some of the areas that have already appeared on the horizon include smart homes, smart cities, smart manufacturing, smart power grids, automated driving solutions, intelligent transportation systems, and environmental monitoring systems. Many more are sure to follow.

But as the new technology transforms our world and the way we live and work, it will also transform the way we think about information technology and the way we build systems. The Internet of Things cannot be approached with conventional thinking, because the conventional approaches simply do not scale. Real Time Web was created as the next-generation platform for the Internet of Things, with a novel outlook on how to build scalable, robust systems.

"The tools we use have a profound (and devious!) influence on our thinking habits, and therefore on our thinking abilities."
– Edsger W. Dijkstra

The Internet of Things

The Internet of Things integrates the physical world with the existing Internet through a vast number of sensors embedded into appliances, vehicles, bodies, and everyday objects. When data from these sensors is integrated into the existing infrastructure, consisting of apps, cloud services, and machine learning, new services are made possible, and new business areas emerge.

Data is gathered from remote sensors, processed in the cloud by advanced machine learning algorithms, and the results are made available to users through apps. The algorithms can leverage historical data collected through the sensors to predict the future more accurately. The sensor network can be augmented with actuators, leading to the emergence of so-called cyber-physical systems, that is, networks that are essentially distributed robots. Applications of the Internet of Things span a wide range, from home automation to environmental monitoring. As yet, the business potential for these applications is largely untapped, but it is likely that the Internet of Things will lead to the creation of whole new technology ecosystems, accompanied by new business models.

Relying on a vast network of sensors is non-trivial from an engineering point of view, for several important reasons. First, all these sensors need life-cycle management. At the lower level, sensor nodes may be self-organizing, but management is still needed when integrating sensors from a whole network. Typical management tasks include configuration management, identity management, ownership management, and security management.

Beyond life-cycle management of sensors, several other challenges exist. The power of the Internet of Things comes from adapting, transforming, analyzing, and integrating sensor data into systems and services. This requires concurrent processing of raw data that originates from a large number of sensors. The sheer volume of data produced by sensors poses serious scalability problems if approached with traditional engineering methods. The traditional approach to handling data from IoT sensors, is to collect and concentrate sensor data and store it in a database. IoT applications can then retrieve data by querying the database via an API and process the data as needed by the application. These traditional solutions lead to problems with efficiency, latency, and throughput and soon become a bottleneck.

Real Time Web – The Next-Generation IoT Platform

Real Time Web was designed to address the shortcomings of the way that IoT systems are built today. It represents a radical departure from conventional methods, while still being completely backwards-compatible with the Internet. In fact, it employs standard web technology to create web-based applications, but the management of sensors and integration of sensor data into these applications is substantially different. Real Time Web was designed to embody several key principles: real-time data transport, extreme scalability, data decoupling, and total interoperability.

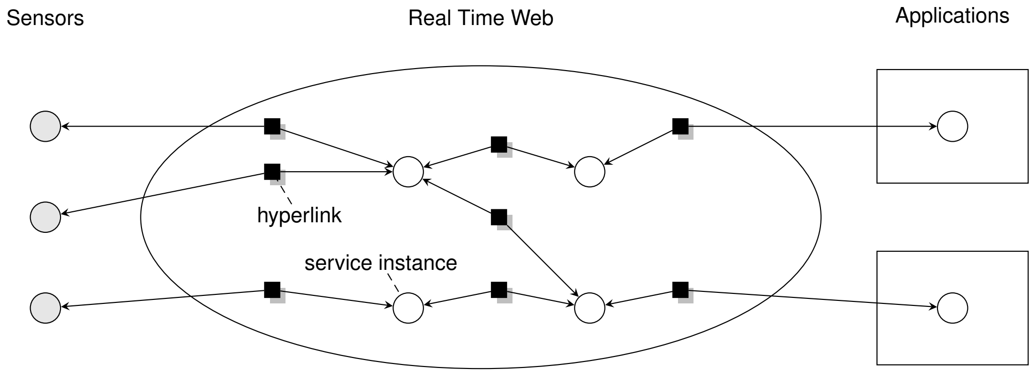
Real-Time Data Transport

Real Time Web is perfectly aligned with the real-time nature of the Internet of Things. Many applications, including smart grids and intelligent transportation systems, require information update in real time, and any significant latency is unacceptable. A particularly crippling aspect of the way that sensor data is typically handled today, is that it follows the "data warehouse" model of traditional applications. In this model raw data is concentrated, processed, persisted, and made available to applications through complicated APIs. This model is typically taken for granted as the standard way to handle and process large amounts of data.

With the massive volume of sensor data that will result from the growth of IoT systems, however, this is no longer a sustainable model. Instead of concentrating and persisting sensor data in large databases, Real Time Web propagates data only where it is needed and when it is needed.

Data Decoupling

Real Time Web applications are based on hyperlinks, not on submitting data via APIs, but it takes a more sophisticated approach to hyperlinks than the WWW. The latter is based on unidirectional hyperlinks, and as it has grown and evolved, the infamous "link rot" phenomenon has become all too familiar. This is caused by the unidirectional nature of the links, that hyperlinks are embedded in the data itself. Real Time Web achieves data decoupling by taking the same approach to hyperlinking as suggested in the original hypertext proposals in Ted Nelson's project Xanadu, that is, by linking data using independent, bidirectional hyperlinks.



These hyperlinks, which follow the subject–predicate–object triple structure of linked data, are separate from the data whose features they describe, so changing hyperlinks is completely independent from revising data. Moreover, the predicate part of the triple can describe functions on data, so the net result is distributed hyperlinks performing transformations on data as it is propagated through the network.

This principle automates the management of hyperlinks between data producers and data consumers. From the point of view of developers, all hyperlinks are localized, and the Real Time Web system automatically handles all routing and processing on the way from data producers to data consumers. When these associations are updated, routing and processing are automatically updated on the fly.

> *"HTML is precisely what we were trying to prevent – ever-breaking links, links going outward only, quotes you can't follow to their origins, no version management, no rights management."*
> *– Ted Nelson*

Extreme Scalability

IoT systems can be large and interconnected as well as highly concurrent. Examples include smart cities, intelligent transportation systems, or smart grids. It is therefore important that such systems can scale linearly when new networks of devices and services are added. This is another case where concentration and storage of sensor data would create unacceptable bottlenecks. Real Time Web emphasizes transfer and processing of data only exactly where and when it is needed. In removing these bottlenecks, it achieves true data distribution and parallelism, and hence provides perfect scalability.

Total Interoperability

Realistically, any IoT framework will need to be backwards compatible with the existing Internet, because one cannot just tear down today's Internet and start all over again. Moreover, it is simply not realistic to expect IoT systems to be built in a homogeneous technology environment. Large, scalable IoT systems will be heterogeneous, built out of many different technologies from a large number of vendors, and all these technologies will need to interoperate seamlessly.

Real Time Web achieves full interoperability along both of these axes. It is based on the Internet and WWW technologies that exists today, so it can co-exist and interoperate with legacy systems. Being protocol agnostic, it can communicate over any protocol used by various IoT technologies, including Dash-7, LoRa, Web Sockets, HTTP and HTTPS, and nodes can also switch protocols dynamically. This design is entirely modular, so for each new protocol that needs to be supported, a driver for it can be developed and installed.

In dynamic networks, where service instances are quickly created and destroyed, the causality of communication between nodes is a problem that is not handled well by conventional approaches. For example, when a new service instance appears and it is relevant to an ongoing communication between other instances, the existing instances have no knowledge of the new instance, and vice versa. Real Time Web solves this problem by automatically notifying the new node to initiate communication with the existing ones, thus supporting a rapidly evolving network of nodes.

Benefits

The immediate benefits of Real Time Web are reduced development and maintenance costs. Due to its emphasis on data decoupling and interoperability, Real Time Web IoT applications can be developed in a matter of days rather than months. And since Real Time Web does not depend on any middleware, there is no extra cost of installation, integration, and system administration.

Applications are future proofed, because there are no APIs, and without APIs, there is no API breakage. Instead of building applications on ever-changing APIs, developers simply create hyperlinks between sensors and services. A hyperlink represents data transfer from a source to a destination and optionally functions to be performed on the data that is transferred. For specific purposes, it is also possible to make a system that creates these hyperlinks automatically, installing new services instantaneously.

Life-cycle management of sensors and service instances is made simple. Vendors that make IoT products containing sensors, simply enroll the products and sensors on their web site. When a user buys an IoT product, the ownership of the sensors in it is transferred to the user. From that point onward, the user has ownership of the sensors and can choose which services to include them in. Ownership of service instances can also be transferred likewise.

Since ownership is preserved across services and networks, it is possible for sensors and service instances to change owner without disrupting ongoing communication between nodes. The Real Time Web system propagates data from sensors via service instances according to policies defined by the owner of each respective object involved in the communication. For example, a user may transfer ownership of a sensor to another user. If the new owner has the same usage policy for this sensor, existing connections to it will continue without disruption. On the other hand, if the new owner prohibits such usage, services depending on this sensor will cease to function.

Users that own things that contain sensors also own the data produced by these sensors. If this data could be useful to others, they may sell it on, effectively creating a market for their own data. Real Time Web puts data ownership in the hands of sensor owners and intrinsically supports trading and licensing of sensor data.

The Real Time Web system can also enforce system consistency policies automatically, enabling advanced security mechanisms similar to those of intrusion detection systems. For example, a consistency rule could define the occurrence of duplicate service instances as an anomalous condition to be handled. Depending on the required security level, the handling policy might be to force a re-authentication of the involved connections, or it could be simply to terminate them. Policies like these can be added or modified without disrupting existing data traffic or services.

Research and Innovation Roadmap

Nornir plans a number of ambitious research and innovation activities in a number of different areas for the next-generation Internet of Things. Real Time Web provides the technological foundation for all these activities, and the work planned here will extend it in several dimensions.

IoT Development Ecosystem

The goal of this research and innovation activity is to design a viable ecosystem for development of IoT applications based on Real Time Web.

Whereas traditional systems revolve around APIs, Real Time Web is based on creating hyperlinks between resources. This leads to a completely different style of development, including changes in engineering practices, processes, and tools. Today, developers of IoT systems are working on large, monolithic codebases involving complex APIs for data access, their work focused on IDEs, libraries, and middleware. Tomorrow's developers will navigate around a vast space of available services and create services and the hyperlinks that connect them. This new working style requires a completely new development ecosystem to support it, consisting of new toolchains and new classes of support services.

A basic support service of this ecosystem is discoverability of data and services. New sensors are constantly being manufactured and deployed by vendors, and new services are constantly being created by other developers. All these resources must be discoverable by developers, but simple discovery is not enough; to build systems, developers also need to understand the semantics of the data that they are linking. This also includes the semantics for metadata, such as usage policies, and requires deep semantic interoperability. A framework standard for the semantics of IoT data and metadata must be defined, as well as a system for associating metadata with sensors and services describing the semantics of their data. Developer tools must support searching the space of sensors and services for any given semantics.

Since the advent of complex distributed systems, it has been a fairly common understanding that some form of viewpoint architecture is needed to be able to communicate and reason about them in a principled way. A viewpoint describes some aspects of the system, while ignoring others. In the traditional development model, where code and middleware are deployed in silos, it would be a nearly impossible task to make tools to generate such viewpoints automatically. But in the Real Time Web model the hyperlinks that describe the system are readily available for analysis, so automated tools are eminently possible. Such tools should allow developers to view IoT systems from various viewpoints, focusing on exactly the aspects required for a particular task, thereby allowing them to understand existing systems faster. These views will cover aspects such as business, functionality, data, distribution, and technology.

The work undertaken in this activity is the definition of the general, conceptual framework for IoT systems development in the Real Time Web system. Within this basic framework, a set of support services will be designed and prototyped, and testing will be done at a realistic scale.

The impact of this activity is that the adoption of next-generation IoT will accelerate, bringing many more new developers aboard. This, in turn, will enable businesses to create IoT systems faster, at reduced cost, thereby boosting revenues while delivering a prolific variety of services that benefit end users.

Low-Level Support for IoT

The goal of this research and innovation activity is to develop the basic network and operating system technology needed to support the next-generation Internet.

When IoT systems are built on the traditional Internet protocols, like TCP and UDP, the real-time performance and the resource efficiency are suboptimal. At some point these inefficiencies will start to affect the scalability and real-time performance of the IoT, so it is important to reduce them to the largest degree. To address these concerns, Nornir has been working on technology to improve the throughput and resource utilization through low-level mechanisms. It is an important goal, however, to maintain backwards compatibility with existing mechanisms, so that the new technology can co-exist and evolve side by side with the existing Internet and operating systems. This poses several practical difficulties, because low-level mechanisms of this nature are normally integrated in the network protocol stack and the operating system kernel.

One such activity is the design of the Service Event Protocol (SEP). The SEP co-exists alongside TCP and UDP, but it offers far better performance characteristics for IoT systems. Since it is an addition to the network stack, however, implementation requires changes in operating system kernels.

Another path is the exploration of distributed operating systems. A distributed operating system provides the same functionality as a traditional OS, that is, processing, storage, and communication, but it transparently distributes this functionality across a large set of physical computers. This model of computation is a perfect fit for Real Time Web, matching its essential properties of flexibility and adaptivity.

The work that will be undertaken in this activity is to develop operating system-level mechanisms and explore their feasibility at a large scale. The IoT systems envisioned at this scale include smart cities and intelligent transportation systems, since these are typically massively concurrent and need real-time operation.

The impact of this activity is that future IoT systems based on this technology will scale better, while also using fewer resources. This leads to the growth of more advanced systems that provide better quality of service at lower energy consumption.

Distributed Artificial Intelligence in IoT

The goal of this research and innovation activity is to extend IoT functionality with intrinsic advanced features, such as artificial intelligence.

Many IoT applications need some form of AI functionality to analyze sensor data. This is particularly important in applications that provide some form of prediction based on a complex pattern of events. For example, in the field of intelligent transportation systems, it would be very valuable to be able predict the risk of congestion based on a combination of time, traffic patterns, and local weather conditions. Another example is in the field of medicine, where wearable sensors could be used to predict certain types of medical conditions in advance.

AI functionality of this kind, is often provided by neural networks. The usual implementation approach is the “data warehouse” model, that is, to concentrate, aggregate and analyze. This would typically be implemented by some sort of cloud service. The big disadvantage of this approach is that it introduces bottlenecks that severely limit the scalability, flexibility, and usefulness of IoT services. It also breaks the fundamental premise of Real Time Web, namely that processing is massively concurrent and distributed. Since Real Time Web is distributed as a network of nodes connected by bidirectional hyperlinks, it is perfectly suitable as an implementation platform for neural networks. Neural networks are processing devices consisting of a set of simple computing nodes interconnected in various types of topologies. The idea is simply to implement these processing nodes and their interconnections as linked data in the Real Time Web system, thereby preserving concurrency and parallelism. This means that the intelligence in processing sensor data is organically embedded in the very structure that propagates this data from producers to consumers.

There are several engineering challenges associated with representing neural networks directly in the Real Time Web system. Unlike normal IoT services, neural networks will not be built by hand, so tools will be needed to automate the process of creating a neural network for a given purpose. Tools are also needed for installation, training, and quality monitoring tailored to each specific case.

The work that will be undertaken in this activity is to design a practical framework for creating and deploying distributed neural networks in the Real Time Web system. This includes tools that handle all the life-cycle tasks of using neural networks, from design and deployment to training and monitoring.

The impact of this activity is that IoT system developers will be able to provide IoT-based prediction and analysis services quickly and reliably, and their customers will benefit from higher quality of service. More businesses will be able to enter the market and provide better value to customers.

Marketplace for Sensor Data

The goal of this research and innovation activity is to define the data trading framework needed to support IoT systems at a massive scale.

Real Time Web allows end users that buy IoT products to take ownership of their sensor data and license it to others. Such licensing requires a marketplace that covers several important aspects of sensor data trade. Designers of IoT services need to obtain data from those sensors that provide the appropriate data for their services, so the first prerequisite for trade is discoverability. In the future there will be a tremendous number of sensors producing data, requiring a principled approach to discovery and selection of data sources for a particular service. Other important aspects of discoverability are quality, pricing, and payment. Sensor data can be of variable quality, depending on the source, and service designers will require some means to acquire sensor data of a quality and reliability they can trust. At the same time they need to consider the price against the quality, and they need efficient mechanisms to pay for sensor data. A sophisticated solution is a marketplace where producers and consumers of sensor data can set up formalized contract conditions and automated contract agreements.

There are, however, several real-world factors that complicate an automated trading system of this kind. First, automated data trade needs to be compatible with legislation, including interoperability across legislative jurisdictions, like national borders. Second, there is also the issue of privacy and security for users that license their sensor data. Many types of sensor data can be of a sensitive nature, particularly when aggregated. It is therefore imperative that producers of sensor data can retain control over exactly what information is shared, providing an assured level of privacy and security. In cases where the identity of the sensor owner is not important, anonymization can be an acceptable solution to the privacy problem.

The work that will be undertaken in this activity is to design a system and a set of associated standards for trading sensor data. The system needs to support an efficient marketplace for sensor data, and it must be able to enforce a broad spectrum of data trading policies.

The impact of this activity is that many new services are made possible, because they can acquire exactly the data they need, of an adequate and controlled quality. This opens up possibilities for many new business areas, and many new businesses will be created, engaging in a flourishing marketplace. At the same time, owners of IoT devices can enter the same marketplace and make money off the data that they deem suitable for sharing.

This document was written
in collaboration with NR.
www.nr.no