

# The Cloud & The Internet of Things



Niels Olof Bouvin

# Overview

- **Cloud computing**
- **The Cloud and the Internet of Things**
- **Fog computing**
- **Putting it all together in the Internet of Things**
  
- **The report template**

# The Cloud?

- **Not *just* marketing-speak for someone else's computer**
  - (though it is that too)

*Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction.*

[NIST Definition]

# On-demand self-service

- **The user of the cloud service can add additional resources (computing, storage, network) as they wish, and when they wish directly through an interface**
- **It might even be possible for the system to add additional resources automatically, depending on the configuration and service plan**
  - e.g., add more servers, if there is a spike in traffic

# Broad access

- **The cloud service is accessible through standard networking protocols**

# Resource pooling

- **The cloud provider can pool their resources, and provide access to their users dynamically**
- **Access is independent of location of the provided machine**
  - though it can be specified, e.g., “give me a server within EU”

# Measured service

- **The user is billed according to the use of resources**
- **This can be continually and dynamically monitored by the cloud provider and user**

# Overview

- Cloud computing
- **The Cloud and the Internet of Things**
- Fog computing
- **Putting it all together in the Internet of Things**
  
- **The report template**



# The Cloud and the Internet of Things

- **A fine match**
- **The Cloud is highly scalable (for a price...), and the IoT has much need for scalability**
  - storage
  - analytics
  - networking
  - access
- **Most commercial IoT systems rely on Cloud services for access**
  - sometimes that is not a good thing, e.g., if the company shuts down or is hacked

# The Cloud for everything?

- **Not necessarily**
  - while the Cloud is great for scalability, it can also become expensive
- **'Edge computing' (also known as P2P) is *free***
- **A combination**
  - Cloud for the backbone, discovery, and the always-on aspects of your system
    - plus elements that people *pay* for
  - P2P for everything else
  - Heavy lifting at the edges, coordination at the center
- **Probably wise, or at least nice, to ensure that your device can function without the Cloud service**

# Overview

- Cloud computing
- The Cloud and the Internet of Things
- **Fog computing**
- **Putting it all together in the Internet of Things**
  
- **The report template**

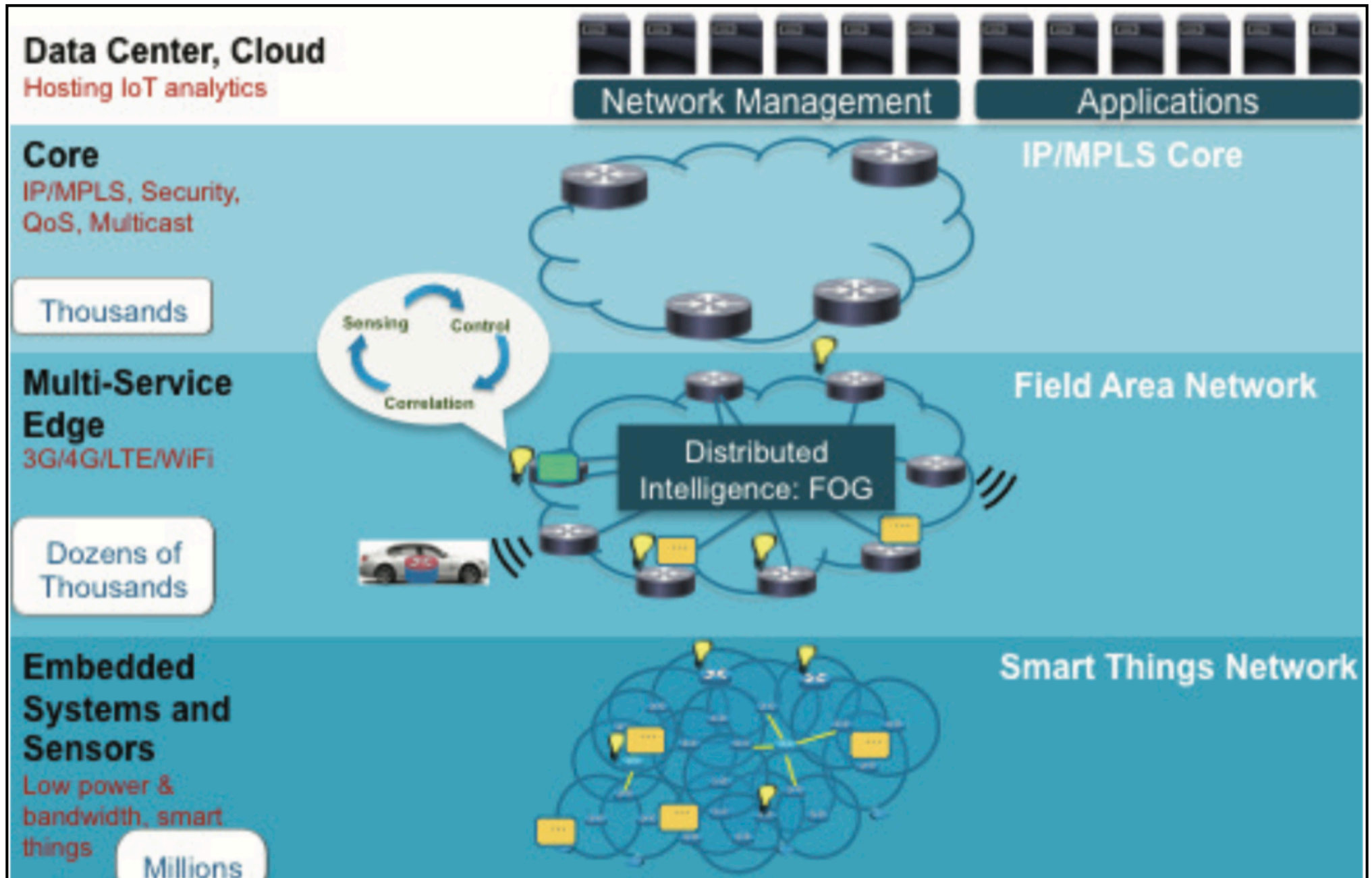
# The Cloud descending

- **Fog computing**
  - originally pioneered by Cisco, a *very* large networking company
- **Like the Cloud, but closer to the user:**

*Fog Computing is a highly virtualized platform that provides compute, storage, and networking services between end devices and traditional Cloud Computing Data Centers, typically, but not exclusively located at the edge of network*

- **It complements, rather than replaces, the Cloud**

# The Fog Computing architecture



# Characteristics of Fog computing

- **All communication is wireless**
  - through WiFi, mobile data, etc
- **Lower latency**
  - the devices are much closer to the user
- **Geographically dispersed**
  - aimed at, e.g., video streaming in cars, or collecting wireless sensor network data
- **What can be processed locally, stays there**
  - this might protect privacy, or could optimise communication to the central cloud
  - (Cisco gets to sell a lot of equipment)

# Overview

- Cloud computing
- The Cloud and the Internet of Things
- Fog computing
- **Putting it all together in the Internet of Things**
  
- **The report template**

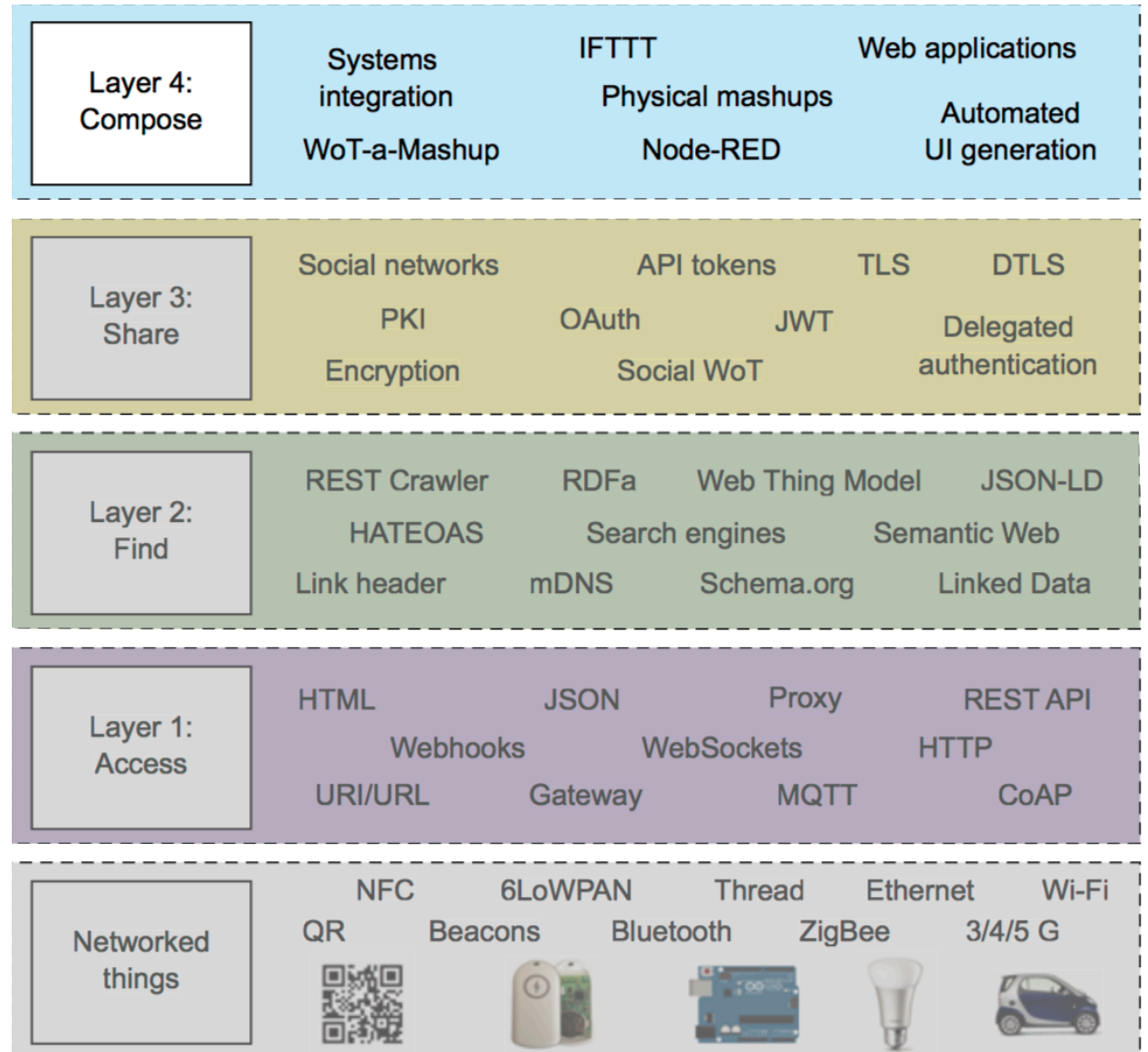
# The Compose layer: mash-ups

- **Combining services to form new ones**

- e.g., combining a temperature sensor with a tweeting service

- **Requirements**

- standard interfaces
- network accessible
- (so well suited for Cloud computing as well as a Web based architecture)





# Standard interfaces

- **If services or devices are to talk together, they must use a shared standard**
  - one example could be the Web Thing Model described by Guinard & Trifa
- **This enables, e.g., automatic generation of UI**

# Node-RED

IBM **Emerging Technology**

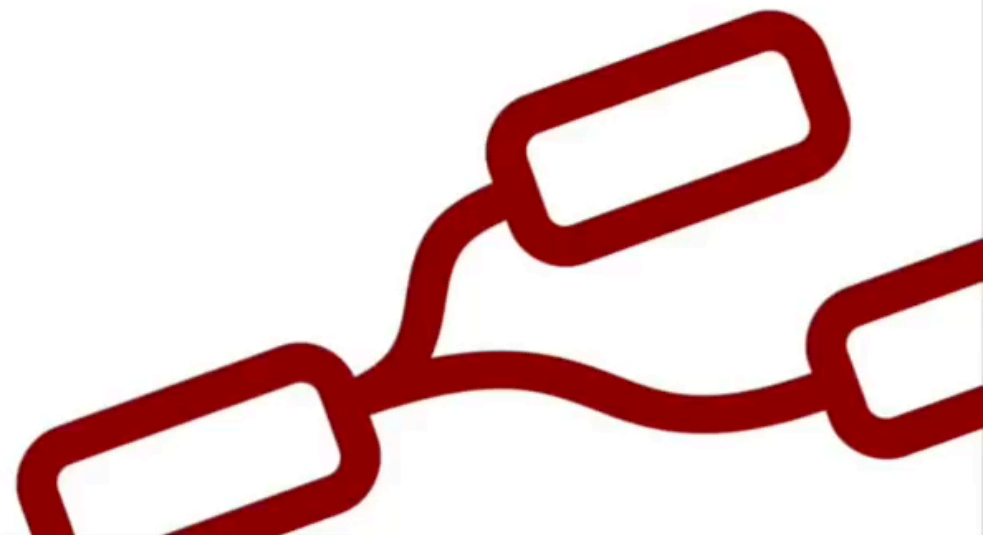
## Node-RED

A visual tool for wiring the Internet of Things

Nick O'Leary

Dave Conway-Jones

**IBM**



- **A Node.js based tool for flow programming on the Web**

# Node-RED flows and nodes

- Node-RED is open source, and there are many, many more flows and components than this
- Runs on everything from RPi to various Cloud services
- Can talk to Arduino

<b>node-red-contrib-osisoft-web-api</b> A node module for connecting to OSISOFT Web API Server v0.0.3 node	<b>node-red-contrib-loxone</b> Connecting the Loxone Miniserver to node-red via Websocket API v0.6.2 node	<b>node-red-contrib-openhab2</b> openhab home automation for node-red v1.0.8 node
<b>Controlling IKEA Tradfri bulb</b> by nygma2004 flow	<b>Temperature</b> by hiapeza flow	<b>node-red-contrib-better-sonos</b> Newer & improved nodes for Sonos devices v0.1.9 node
<b>asdsa</b> by kandrianov flow	<b>node-red-contrib-diode</b> A simple node to isolate the output of a node or function to that further processing cannot affect the original. v1.0.1 node	<b>node-red-contrib-grove</b> A simple interface to Grovestreams graphing site v1.0.9 node
<b>node-red-contrib-esplogin</b> This node is specifically designed to work with tech.scargill.net home control software. v1.3.6 node	<b>node-red-contrib-timeout</b> A simple timeout node which can also be used as a toggle for testing. v1.0.7 node	<b>node-red-contrib-bigtimer</b> The ultimate Node-Red Timer with dusk, dawn (and variations), months, days, manual override, schedule pause, random or fixed v1.7.8 node
<b>node-red-contrib-watson-ucg-custom</b> A collection of Node-RED nodes for IBM Watson services v0.5.5 34 node	<b>Live tweet and sentiment analysis with mongodb</b> by 0x69676f72 flow	<b>Pointing on the map the collected tweets and specifying its sentiment using IBM Watson.</b> by 0x69676f72 flow
<b>Deploys each message and tweeting collected search.</b> by 0x69676f72 flow	<b>node-red-contrib-opentherm</b> OpenTherm interface for Node-RED v0.1.1 55 node	<b>node-red-contrib-dynamorse-core</b> Setup and support for Streampunk Media's Node-RED nodes v0.1.14 288 node
<b>node-red-dashboard</b> A set of dashboard nodes for Node-RED v2.6.1 6045 node	<b>node-red-contrib-discovery-insert</b> A node for inserting documents into the watson discovery service. v0.1.7 325 node	<b>node-red-contrib-huemagic</b> Philips Hue node to control lights, groups, scenes, motion sensors, temperature sensors and Lux sensors using Node-RED. v1.1.9 155 node

# Summary

- **The Cloud offers a rich set of not only technologies, but also services that can be combined with Things to provide a richer and more seamless experience**
- **The more standardised the interfaces between the components are, the easier and stronger the interaction can be**
- **With the right kind of interfaces, building a mash-up can be (nearly) as easy as drawing lines between blocks**

# Overview

- Cloud computing
- The Cloud and the Internet of Things
- Fog computing
- Putting it all together in the Internet of Things
- **The report template**

# The report template

- It's *my* job to read your reports, so I get to decide how they are structured and what they look like
- A L<sup>A</sup>T<sub>E</sub>X template based on the work by André Miede, who in turn was inspired by the excellent "*The Elements of Typographic Style*" by Robert Bringhurst
- I have used it for a couple of years with my master thesis students, but it works just as well for reports

# Getting the template

```
git clone https://users-cs.au.dk/bouvin/git/thesis.git
```

- **As well as a link to a PDF and zip-file on the frontpage**
- **Link to ShareLaTeX version forthcoming today or tomorrow, baring surprises**

# The overall structure

- **Part I**
  - the proper structure of a thesis (written by yours truly)
- **Part II & Part III**
  - formatting instructions and suggestions (by André Miede)



# The proper structure of a report

- **Introduction**
- **Related work**
- **Analysis**
- **Design**
- **Implementation**
- **Evaluation**
- **Conclusion**

# Introduction

- **What is it? Why is it interesting?**
- **What is the use case for it?**
- **What are the central questions/hypotheses about it?**
- **How can those questions be tested?**

# Related work

- Scientific and technical works that are *relevant* and *contribute* to the analysis of the problem domain
- Identify central aspects in the literature
- Use those aspects to systematise your write-up of each article or system
- Discuss *only* the work of others—your own work comes later!

# Analysis

- **How can you use the Related Work in the context of your project?**
- **Which aspects are desirable, and which challenges are yet to be met?**
- **What is it you wish to do better or differently?**

# Design

- **The crystallisation of the analysis**
- **The abstract design of your system**
- **Can involve more than what you will end up building**

# Implementation

- The parts of the design necessary to answer the questions in the introduction implemented
- The purpose of your implementation is *strictly* to evaluate your hypotheses
  - no more, no less

# Evaluation

- **Testing your implementation to investigate your questions or hypotheses**
- **Make your experiments as clear and focused as possible**
- **Compare expectations and results, and discuss**

# Conclusion

- **Wrapping up your work**
- **Ideally, you should be able to read the introduction and the conclusion, and make sense of what you have done**
- **Revisit the questions and hypotheses from the introduction? Have they been answered adequately?**
- **What are still open questions?**
- **What are the consequences of your results?**