Preliminary Title

Preliminary Description

Group Number: 107

Joar Heimonen, Christian Vu, Naly Keli

`contact@joar.me`

`chvu002@student.kristiania.no`

`nake002@student.kristiania.no`

May 11, 2025

Abstract

Preliminary Abstract

Contents

1 Introduction

There are many paradigms of commercial sensor management and monitoring. Organizations use among other things PLC (programmable Logic Controllers), IoT devices and SCADA (Supervisory Control and Data Acquisition) systems to manage and monitor their sensors. For commercial use some of these alternatives are more popular than others. There are also a large amount of different higher level protocols like MQTT, HTTP and SNMP that can be used to manage and monitor sensors. We propose using the NETCONF protocol with YANG sensor models for management. This work will be done in collaboration with Lightside Instruments AS.

This document will cover the following three topics:

- **Work Methodology:** An indept analysis of the knowledge base around work methods like Scrum, Kanban, and Waterfall. With a focus on how our work methodology differs from these.
- **NETCONF and YANG sensor management**: A qualitative analysis of the NETCONF and YANG protocols and how they can be used to manage sensors and a description of our work with Lightside Instruments AS on NETCONF and YANG sensor management.
- **NETCONF Security**: A qualitative analysis of the security aspects of the NETCONF protocol.

2 Lightside Instruments AS

Lightside Instruments is a company specializing in developing instruments with model based network management for use in networking, network interconnect testing and telemetry. They design their instruments with YANG (RFC7950) [6] models and NETCONF (RFC6241) [11] protocol. The instruments are based on IETF standards and drafts, and are implemented with software tools available in Debian, programmable logic and open hardware [22].

3 Technical background

3.1 NETCONF and YANG

NETCONF [11] is a model based Network Configuration Protocol. Each NETCONF device presents the aquiring device with a YANG [6] data model consisting of the device state and parameters. Each data model has a set of constraints making them error correcting.

3.2 Yuma123

Yuma123 [34] is an open source NETCONF and YANG implementation. Yuma123 is available as a Debian package and is maintained by Lightside Instruments AS.

3.3 NodeJS

Node.js [26] is a JavaScript runtime built on Chrome's V8 JavaScript engine. It is used to build JavaScript applications that can run independently of a web browser. Node.js is popular for building server-side applications as it is relatively performant and it allows developers to work with the same language on both the client and server.

3.4 Node-Yuma123

Node-Yuma123 [25] is a NodeJS package that implements a set of Yuma123 bindings. It allows for efficient NETCONF and YANG development in NodeJS.

3.5 Node-RED

Node-RED [23] is an open source low code programming tool for event driven applications. It is developed by IBM and is based on Node.js [26]. Node-RED is used to connect hardware devices and APIs through a visual programming interface.

3.6 Grafana

Grafana [16] is an open source data visualization tool. It is used to visualize arbitrary data from different data sources.

3.7 XML and XPATH

XML stands for Extensible Markup Language [12]. It is a markup language that is used to encode data in a format that is both human and machine readable. XPATH [37] is a language for navigating XML documents. An XPATH expression allows for the selection of arbitrary nodes in an XML document.

3.8 Scrum

Scrum [18] is a framework for agile [5] software development. The term Scrum is derived from the game of rugby, where a scrum is a way of restarting play after a minor infringement [31].

3.9 Kanban

Kanban [20] is a framework for agile software development. The term Kanban is derived from the Japanese word for ”signboard” or ”billboard” [21].

3.10 Waterfall

Waterfall [36] is a framework for software development. The waterfall model is a linear and sequential approach to software development.

3.11 Extreme Programming

Extreme Programming [13] is a framework for agile software development. Extreme Programming takes the best practices of software development and takes them to the extreme, hence the name.

4 Work Methodology

4.1 Research Question

This review examines the claims that Scrum, Kanban, Waterfall, Extreme Programming and DevOps increases worker productivity substantiated by empirical evidence.

4.2 Scoping Review

4.2.1 Search strategy

The following is our search strategy for the scoping review. We will be searching for quantitative studies on the efficiency of the following work methodologies:

- Scrum
- Kanban
- Waterfall
- Extreme Programming
- DevOps

We will be searching the following databases:

- IEEE Xplore [19]
- ACM Digital Library [3]
- Google Scholar [15] (Meta database)

We will also be searching the following industry websites:

- Agile Alliance [4]
- Scrum.org [18]
- DevOps Institute [27]
- Scrum Alliance [32]

Our search will consists of a set of primary and secondary keywords. The primary keywords are:

- Scrum
- Kanban
- Waterfall

- Extreme Programming
- DevOps

The secondary keywords are:

- Effectiveness
- Efficiency
- Productivity
- Performance
- Success
- Failure

The search will be done using the following search string:

```
(Scrum OR Kanban OR Waterfall OR "Extreme Programming" OR DevOps)
AND
(Effectiveness OR Efficiency OR Productivity OR Performance OR Success OR Failure)
```

4.2.2 Exclusion Criteria

The systematic review will include articles meeting the following criteria:

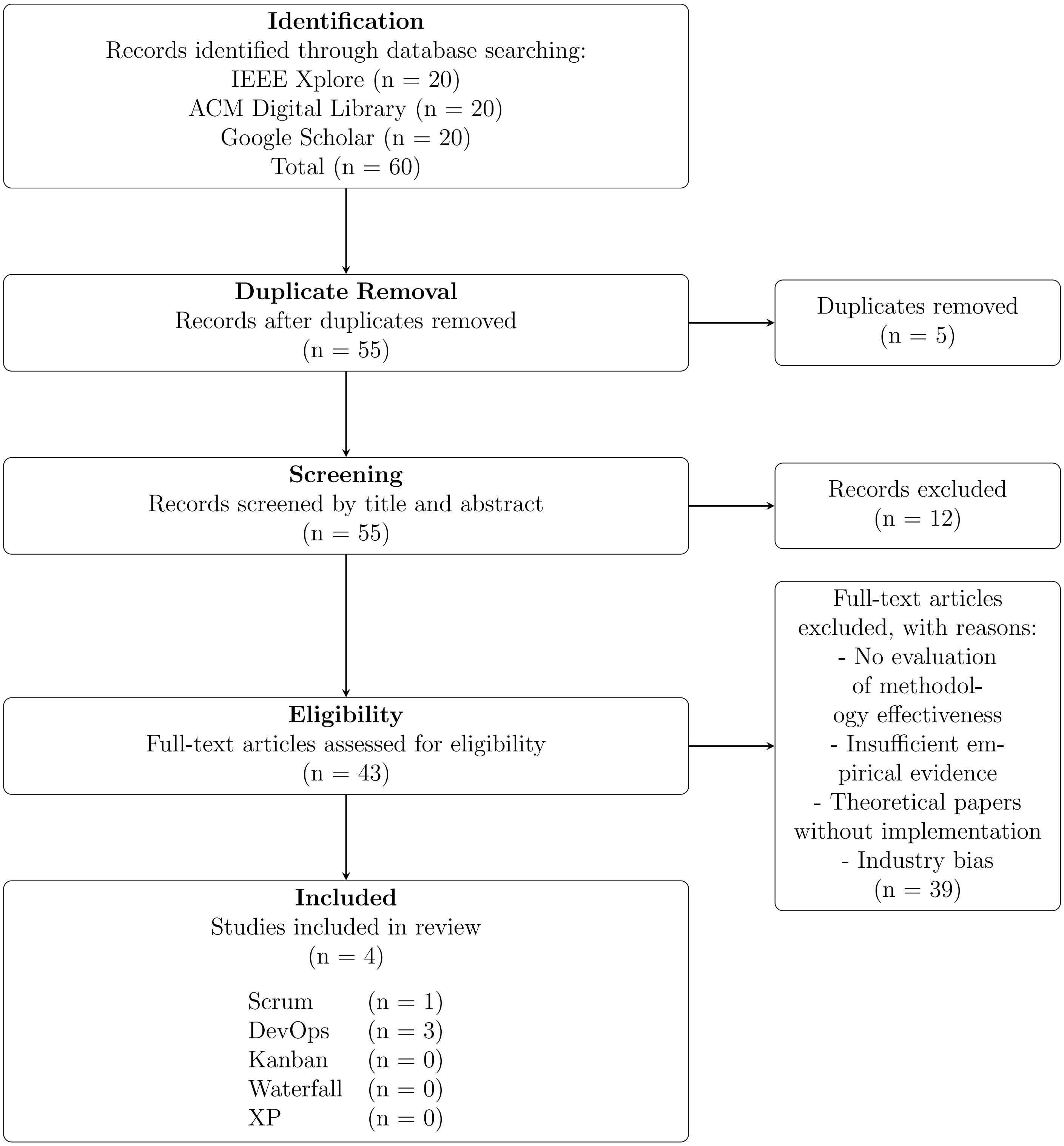
- Published after January 1, 2020
- Published in English
- Relevant to the research question
- Empirical evidence
- Quantitative studies
- One of the 20 first results from each database
- Evaluating the effectiveness of the following methodologies:
  - Scrum
  - Kanban
  - Waterfall
  - Extreme Programming
  - DevOps

4.2.3 Result

After applying the exclusion criteria to a set of 60 articles, we discovered that 4 of them were duplicates. The 56 remaining articles were screened by title and abstract, resulting in 12 articles being excluded. The 44 remaining articles were assessed for eligibility, resulting in 33 articles being excluded. The 11 remaining articles were included in the review.

4.2.3.1 PRISMA flow diagram

Figure 1 shows the PRISMA [30] flow diagram for the scoping review. The PRISMA flow diagram is a standardized way of reporting the results of a scoping review.



Identification
Records identified through database searching:
IEEE Xplore (n = 20)
ACM Digital Library (n = 20)
Google Scholar (n = 20)
Total (n = 60)
Duplicate Removal
Records after duplicates removed (n = 55)
Duplicates removed (n = 5)
Screening
Records screened by title and abstract (n = 55)
Records excluded (n = 12)
Eligibility
Full-text articles assessed for eligibility (n = 43)
Full-text articles excluded, with reasons:
- No evaluation of methodology effectiveness
- Insufficient empirical evidence
- Theoretical papers without implementation
- Industry bias
(n = 39)
Included
Studies included in review (n = 4)
Scrum (n = 1)
DevOps (n = 3)
Kanban (n = 0)
Waterfall (n = 0)
XP (n = 0)

Figure 1: PRISMA flow diagram for scoping review of software development methodologies

4.3 Scrum

Scrum is a framework for agile software development. The term Scrum is derived from the game of rugby, where a scrum is a way of restarting play after a minor infringement [31]. The use of the term Scrum in software development was first introduced by Takeuchi and Nonaka in 1986 in a paper titled "The New New Product Development Game" [33]. In the paper, the authors argue for a new approach to product development where the different stages of development are overlapped, rather than sequentially executed in a "pass the baton" fashion. This differs from the then popular NASA type PPP (Phased Project Planning) model [29].

Modern Scrum development consists of a set of sprints, these sprints consists of a pre-defined set of tasks that are to be completed in the pre-defined sprint time frame. Each task or "story" is assigned an arbitrary number of points that represents the complexity of the task. The sprint is the completed when there a no more points to be completed or the time frame is up. There are many modern flavors of Scrum, like Accenture's [2] Autoscrum which is a scrum framework that was first introduced in the talk "AGILE TRANSFORMATION? FOR COMPLEX SYSTEMS? ...NO WAY!" by Brehm [7] as can be seen in *Figure 2*. Or the scaled agile framework (SAFe) [14] which is a framework developed by Scaled Agile Incorporated which introduces SAFe Scrum see *Figure 3*. Or the Deloitte's [9] "The Agile Landscape v3" that consists of all the different frameworks and methods used for project management. See the Scrum section in *Figure 4*.

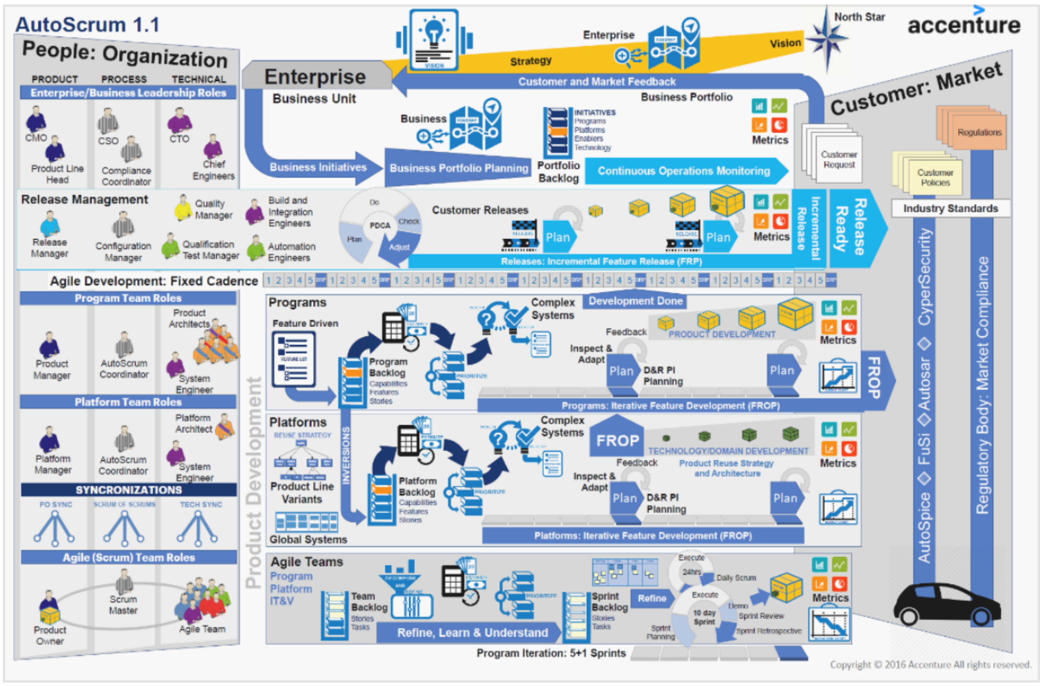


Figure 2: Excerpt from the presentation "AGILE TRANSFORMATION? FOR COMPLEX SYSTEMS? ...NO WAY!" by Brehm [7] showing the Autoscrum framework.

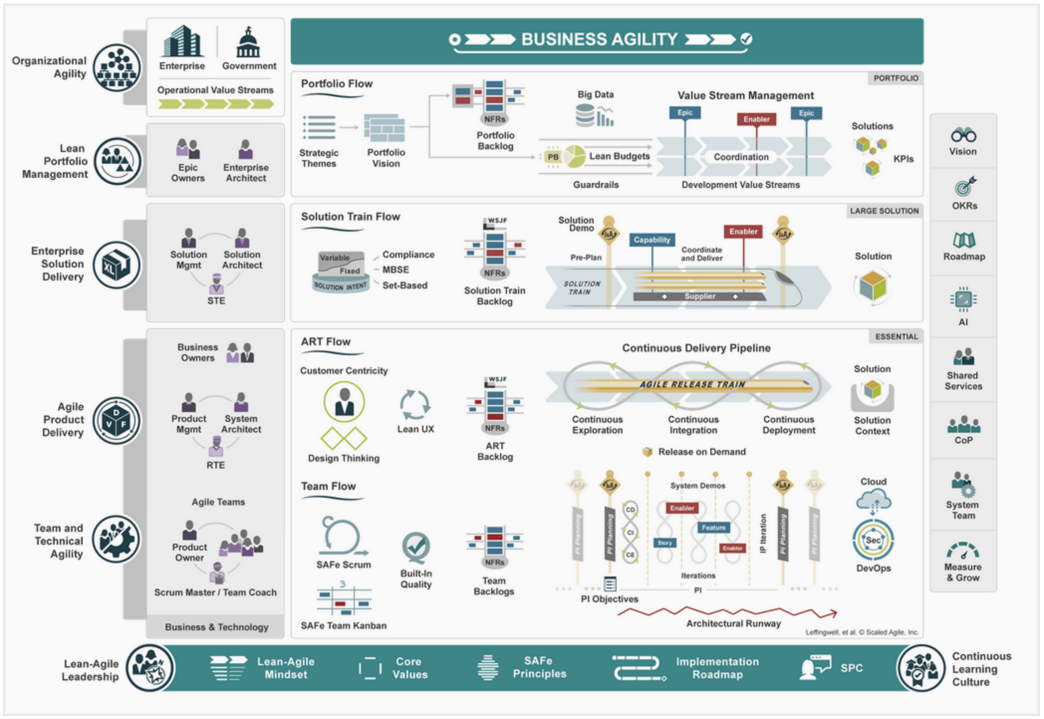


Figure 3: The Scaled Agile Framework (SAFe) [14].

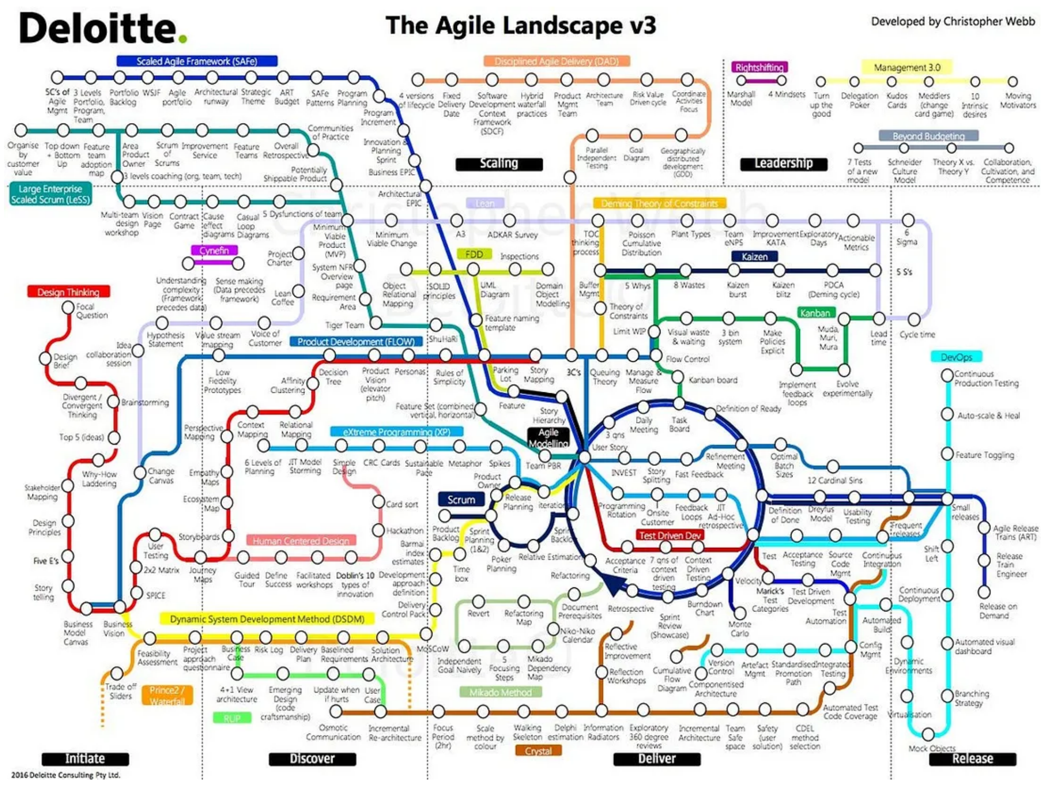


Figure 4: The Agile Landscape v3 [9] showing the different frameworks and methods used for project management.

4.3.1 Evidentiary foundation

With only one article "A Theory of Scrum Team Effectiveness" by Verwijs and Russo [35] we can conclude that there is no empirical evidence that Scrum increases worker productivity or efficiency. Whilst the findings of the article seems promising, and the article is thorough, there simply is not enough research on the topic to comment on the evidentiary foundation of Scrum, this is something taken in too account by the authors of the article. Only a handful of other articles have been published on the topic, but most of them fall outside the exclusion criteria of the scoping review as they have been published before 2020.

4.4 Kanban

Kanban is a framework for agile software development. The term Kanban is derived from the Japanese word for "signboard" or "billboard" [21]. A Kanban board is a visual representation of the tasks that need to be completed, the board consists of multiple columns that represent the different stages of the development process. As a task is worked on it traverses the columns of the board until it is completed. This provides a visual representation of the state of the project and allows for easy identification of bottlenecks in the process. Kanban is a much more flexible and lean approach to project management than Scrum, as it follows closer to the agile manifesto's principles of flexibility and adaptability [24].

4.4.1 Evidentiary foundation

With no articles found in the scoping review we can conclude that there is no empirical evidence that Kanban increases worker productivity or efficiency.

4.5 Waterfall

The waterfall [36] is a linear and sequential approach to software development. Sets of tasks are grouped into phases, where each phase must be completed before the next phase can begin. This is reminiscent of the NASA type PPP (Phased Project Planning) model [29]. The waterfall model mirrors the traditional problem solving process, of breaking a problem down into a set of smaller problems, and solving each of the smaller problems in a sequential manner. This is something that is found at the core of all project management methodologies.

4.5.1 Evidentiary foundation

With no articles found in the scoping review we can conclude that there is no empirical evidence that Waterfall increases worker productivity or efficiency.

4.6 Extreme Programming

Extreme Programming [13] is a framework for agile software development. Extreme Programming takes the best practices of software development and takes them to the extreme, hence the name. A core part of Extreme Programming is the use of pair programming [28], where two developers work together on the same code.

4.6.1 Evidentiary foundation

With no articles found in the scoping review we can conclude that there is no empirical evidence that Extreme Programming increases worker productivity or efficiency.

4.7 DevOps

DevOps is the combination of development and operations, it is the practice of combining software development and IT operations. This keeps the developers close to the day to day operations of the software they are developing. A result of developer operations is the use of automated continues integration and continues deployment (CI/CD) [8] systems as the responsibility of the deployment falls on the developers. This methodology fosters early error detection and correction.

4.7.1 Evidentiary foundation

DevOps and Research Assessment (DORA) [10] is a research department at Google that focuses on researching assessment methods for DevOps. They publish a yearly report on the state of DevOps. Other than DORA independent research into the topic is scarce and whilst research is being done there is currently not a strong enough evidentiary foundation to make any claims about the effectiveness of DevOps.

4.8 Conclusion

The scoping review has made it clear that there is a lack of research into the effectiveness of different work methodologies. The industry seems focused on researching performance metrics for each methodology, rather than the effectiveness of the methodology itself.

4.9 Our Work Methodology

At Lightside Instruments we work with both hardware and software in a small team. This workflows lends a great deal of flexibility to our work. With most engineers doubling as both hardware and software engineers, we have found it cumbersome to use a strict work methodology like most agile frameworks. Due to the small size of our team we have adapted a DevOps like approach that is tailored for our needs. We have a set of tasks that are defined in a git repository. After a task is completed a pull request is submitted and the task is marked as pending review. When the request is approved the task is marked as completed and the code is merged into the main branch. We not only use automated continues integration and continues deployment (CI/CD) systems for our software, but also for our hardware. We have developed a set of workflows that are used to automatically generate the production files for our hardware.

Whilst there is no empirical evidence or any research suggesting that our work methodology is more effective than any of the other methodologies, we have found it to be a good fit for our needs. And we feel that it adheres close the principles set forth by the agile manifesto [24].

5 NETCONF and YANG sensor management

Hardware management is an essential part of administrating a larger network. Together with Lightside Instruments AS we have developed open source tools for YANG and NETCONF aimed at hardware sensor management. We have built a reference implementation of a NETCONF system that tackles this issue. Our reference implementation consists two NETCONF temperature

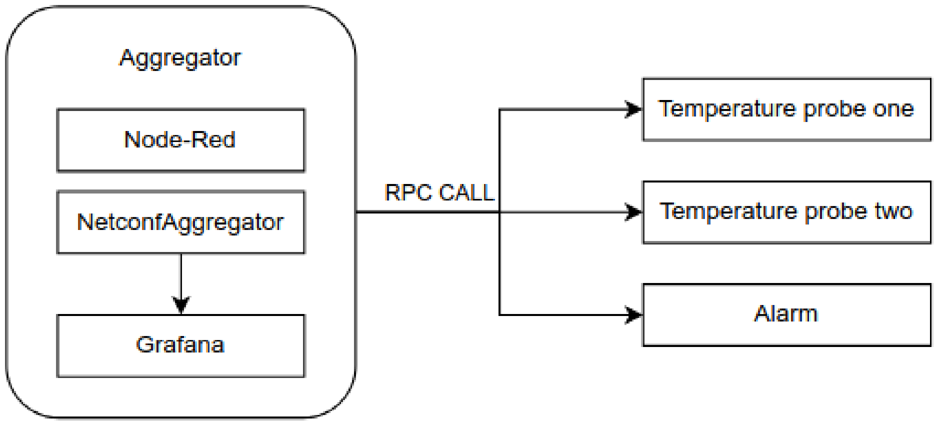


Figure 5: Architecture of the NETCONF and YANG sensor management system.

probes, a data aggregator, a data visualization tool and a tool for quick management of NETCONF devices.

The following sections will describe the project and how each of the components are implemented. An illustration of the architecture of the project can be seen in *Figure 5*.

5.1 NETCONF

NETCONF [11] is a mode based Network Configuration Protocol implemented by the IETF (Internet Engineering Task Force) in the RFC (Request for Comments) 6241 [11].

5.2 YANG Model

The YANG model is a model used to describe the state and actions of a NETCONF device. The Internet Engineering Task Force (IETF) has developed a set of standard YANG models for NETCONF devices. For the purposes of this project we will not be using the standard YANG models, but instead we will be using a custom YANG model developed by Lightside Instruments AS that only describes the state of thermometers. See *Figure 6* for the YANG model.

```
module lsi-thermometers {
  yang-version 1.1;
  namespace "urn:lsi:params:xml:ns:yang:thermometers";
  prefix thermometers;

  organization  "Lightside Instruments AS";

  description
    "Thermometers monitoring module.";

  revision 2022-07-25 {
    description
      "Initial version.";
  }

  container thermometers {
    config false;
    list thermometer {
      key "name";
      leaf name {
        type string;
      }
      leaf value {
        description
          "Temperature in degrees Celsius multiplied by 100.";
        type int32 {
          range "-27315..max";
        }
      }
    }
  }
}
```

Figure 6: YANG model for thermometer management

5.3 Node-RED

Node-RED is a low code programming tool for event driven applications. It makes it possible to create arbitrary flows that function as a compatibility layer between different systems and protocols. The low code nature of Node-RED makes arbitrary system integration accessible even for the lay person.

5.3.1 Red-Netconf

Red-Netconf [1] is a Node-RED plugin that implements the following two nodes:

- **Netconf Session**: This node is used to create a NETCONF session with a NETCONF device.
- **Netconf Yangcli**: This node is used to send NETCONF commands to a NETCONF device using yangcli commands.

Using these two nodes we are able to create Node-RED flows that can manage NETCONF devices.

5.3.1.1 Temperature alert

As an example of how the Red-Netconf nodes can be used we created a Node-RED reference flow that collects data from a thermometer and switches on an LED when the temperature is above 25 degrees Celsius, this can be seen in *Figure 7*.

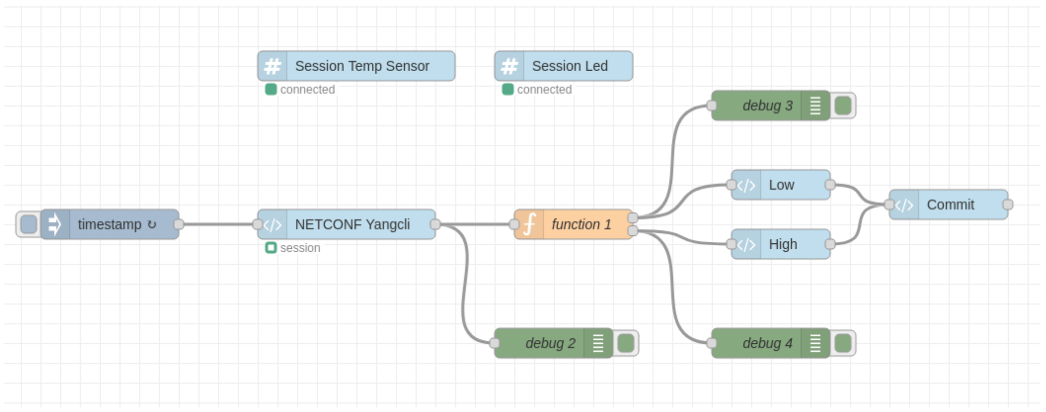


Figure 7: Node-RED flow using Red-Netconf nodes that monitors a temperature sensor and switches on an LED when the temperature is above 25 degrees Celsius.

5.3.2 Node-Yuma123

Yuma123 is an implementation of the NETCONF standard that is written in C. It consists of a NETCONF server written in C, an RPC (Remote Procedure Call) client and a YANGCLI interpreter. The YANGCLI interpreter is a command line tool that can be used to interact with NETCONF devices using YANG models and YANGCLI commands. YANGCLI translates the YANGCLI commands into NETCONF RPC calls and sends them to the NETCONF device. Yuma123 also presents an API that can be used to interact with the NETCONF server called libyuma-dev.

Node-Yuma123 is a set of NodeJS bindings for the Yuma123 API. It is written in C++ and created to resemble the original C API as closely as possible. Node-Yuma123 also implements some new functions like asynchronous connections. Implementing this was quite a challenge as the libyuma-dev API is not thread safe. Instead of running each connection in a new thread we stack the instructions using libuv [**libuv**] and run them in the main thread. This resolves most of our threading issues, but some issues still remain. These issues are related to how libyuma-dev allocates memory.

5.3.2.1 easyNetconf

C is a low level functional programming language, there are many implementations of the C compiler. All of these implementations follow the C standard set forth by the American National Standards Institute (ANSI) in the ANSI C standard. Due to JavaScipts object-oriented nature we decided to create a wrapper around the node-Yuma123 library. This wrapper allows us to use the node-Yuma123 library in an object-oriented way.

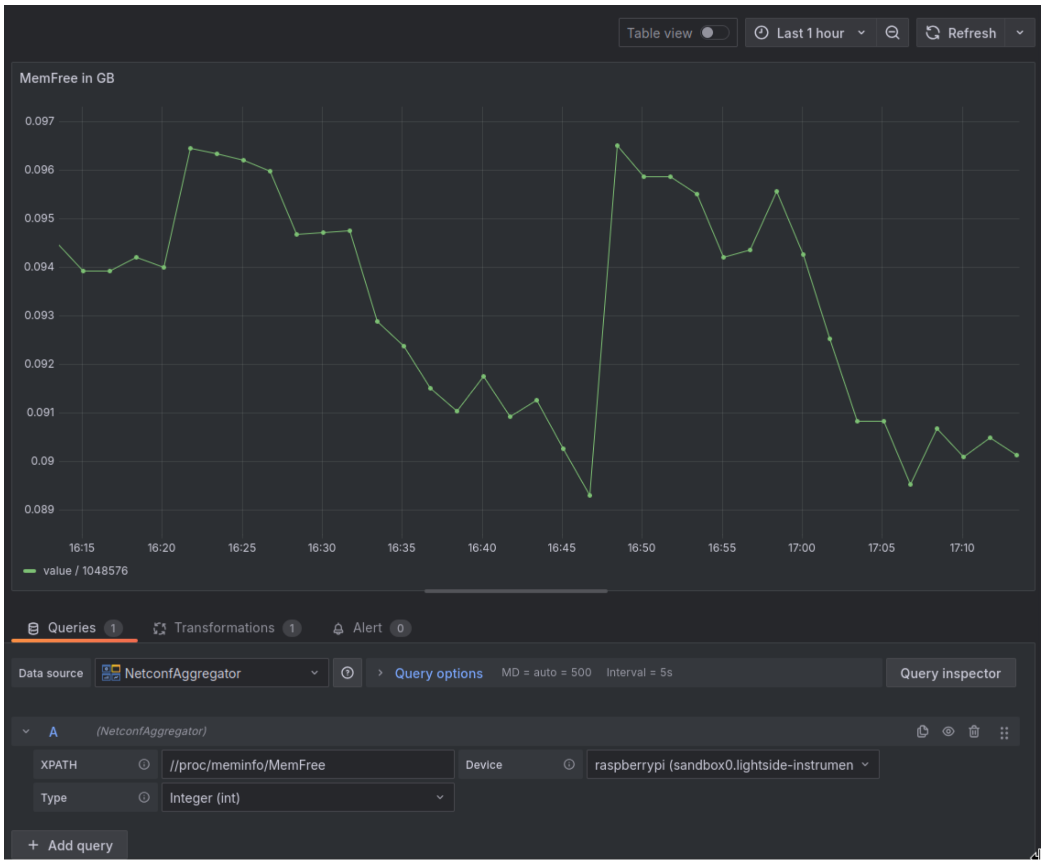
5.4 Grafana

Grafana is an open source data visualization tool. Since its inception ins 2014 it has become the industry standard for data visualization. It is used to visualize arbitrary data from different data sources. We have developed a solution for fetching and visualizing arbitrary NETCONF data using Grafana. We call this solution NetconfAggregator.

5.4.1 NetconfAggregator

NetconfAggregator is a NodeJS package that uses the easyNetconf wrapper [17] to aggregate NETCONF data and store it in a postgresql database. The data is stored in the form of a time series consisting of an ID, a timestamp and the raw XML data from the NETCONF device. This allows the data to be extracted and visualized using XPATH queries. An example of this can be seen in *Figure 8*. This example shows how the query *"//proc/meminfo/MemFree"* can be used to create a graph of the memory usage on a raspberry pi trough the NetconfAggregator datasource plugin in Grafana.

Figure 9 shows the architecture of the NetconfAggregator, consisting of a PostgreSQL database, a NetconfAggregator instance and a Grafana instance running the NetconfAggregator datasource plugin. The NetconfAggregator gets the state of each NETCONF device using the YangCLI command *"xget /"* which fetches the state of the device. The response is then stored in the database. When the NetconfAggregator receives a query from the datasource plugin it applies the specified XPATH to the relevant data-range in the database and returns the result to the datasource plugin.



Data source: NetconfAggregator
MD = auto = 500 Interval = 5s
A (NetconfAggregator)
XPATH: //proc/meminfo/MemFree
Device: raspberrypi (sandbox0.lightside-instrumen
Type: Integer (int)

Figure 8: Querying the memory free data on a raspberry pi using XPATH and the NetconfAggregator plugin.

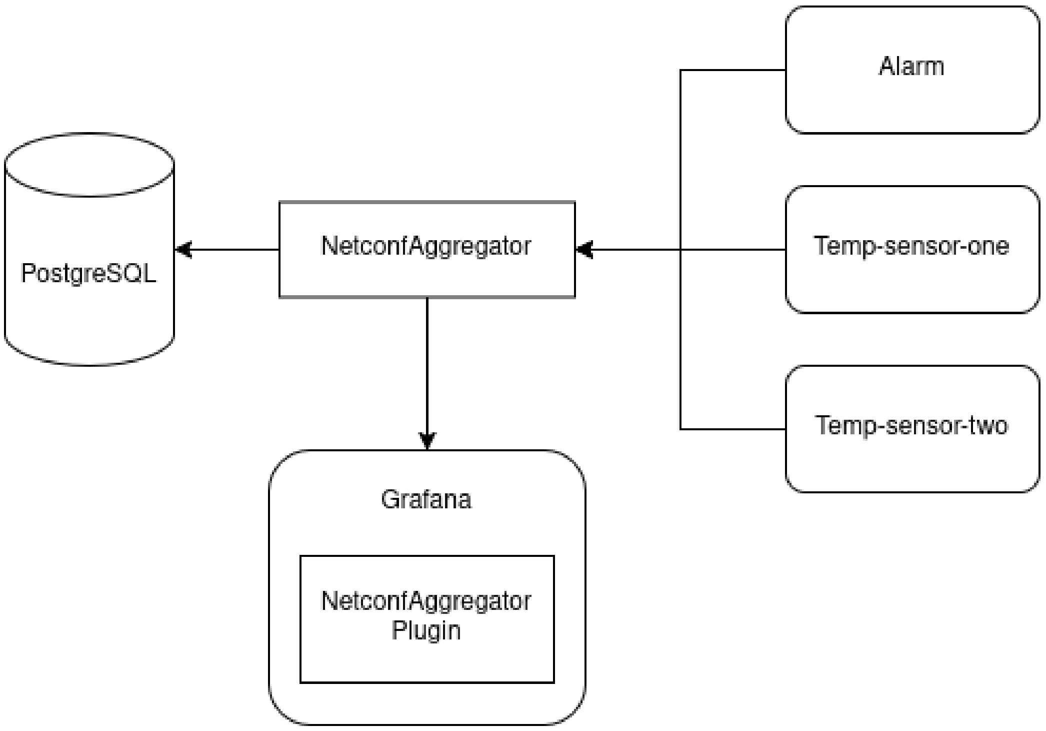


Figure 9: NetconfAggregator architecture.

We quickly discovered that processing XPATHS is processor intensive. Especially when the data set is large. Grafana also has a tendency to send all the queries belonging to a dashboard at once therefore, we decided to implement a caching system, where the results of the XPATH queries are cached in memory, stored in a hash table. However, this did not work as expected, as the queries were arriving at the same time the responses did not have time to reach the cache before the next identical query arrived. To solve this we implemented a queue system, we call it the event queue. When a query goes to processing it is simultaneously added to the event queue. Future identical queries are delayed until the processing of the first query is completed. A delay is added to each request that the server receives. This delay is based on the number of queries that are currently being processed, and thus makes sure that no two identical queries are processed at the same time. An illustration of how requests are processed can be seen in *Figure 10*.

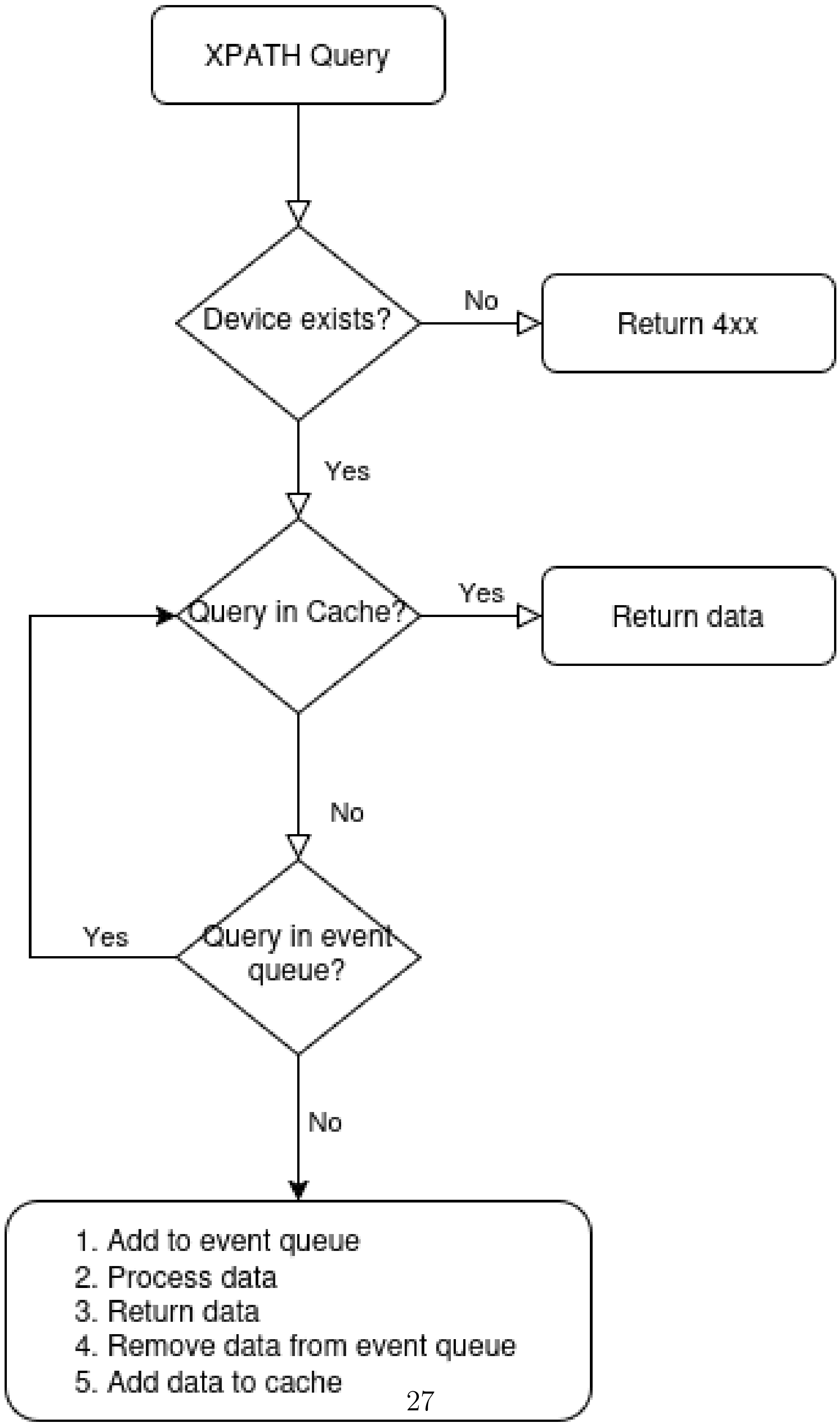


Figure 10: XPATH query processing.

5.5 Decentralized monitoring

We believe that the power of NETCONF lies in its ability to be used in a decentralized manner. This means that multiple aggregators can be used to manage the same NETCONF devices. The example in *Figure 11* shows two servers each containing a node-red instance, a NetconfAggregator instance and a Grafana instance. The node-red instance and the NetconfAggregator functions completely independently of each other. Thus making the monitoring system completely separate from the management system. The example consists of a normal networking setup consisting of three NETCONF enabled switches, a NETCONF enabled temperature sensor and a NETCONF enabled alarm. This architecture allows for arbitrary control and monitoring of the NETCONF devices using the NetconfAggregator and Node-RED.

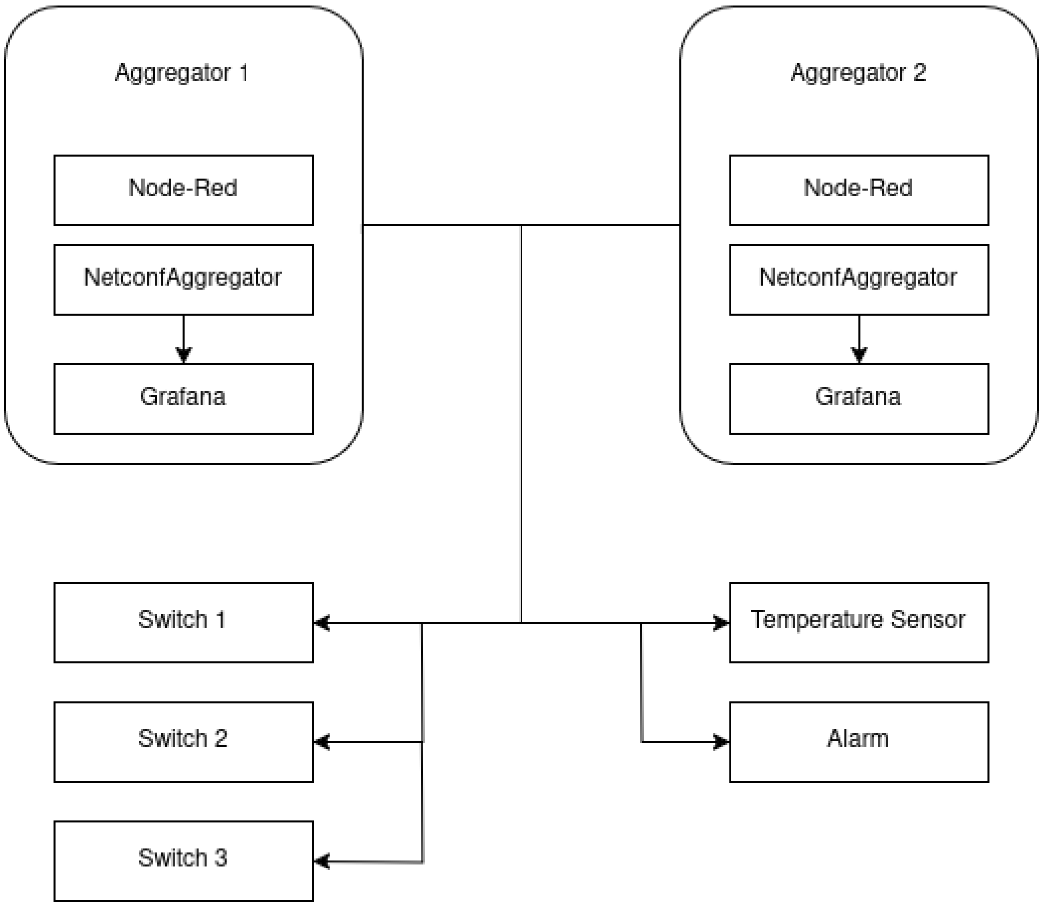


Figure 11: Decentralized monitoring of NETCONF devices using Grafana, NetconfAggregator and Node-RED.

6 NETCONF Security

The security of Remote Procedure Calls (RPC) is an important part of any NETCONF setup. NETCONF is developed in a manner that allows it to function over any transport layer, where the most commonly used is secure shell (SSH). By utilizing SSH the protocol is able to outsource the security of the transport layer allowing the protocol to focus on the management of network devices instead of the security of the transport layer. This type of compartmentalization is common when it comes to IETF standards. Other notable examples of this are the Internet Protocol Security (IPsec)

[**RFC4301**] and the Transport Layer Security (TLS) [**RFC5246**] protocols.

6.1 SSH hardening

A secure shell (SSH) is only as good as the configuration of the SSH server. The process of securing an SSH server is called "hardening". There are many ways to harden an SSH server. The following are some of the more common practices. Asymmetric encryption is one of the most common ways of securing an SSH server. This involves every user generating a public/private key pair, the public key is then passed to the server and the private key is kept on the client. The server can then use the public key to encrypt messages that can only be decrypted by the private key. A central step in initializing an SSH connection is the key exchange (KEX) between the client and server. This is done through the Elliptic-curve Diffie-Hellman (ECDH) KEX protocol. As seen in *Figure 12* the exchange starts by the two parties in the key exchange selecting a large common prime, in this case $p = 7$, after this a generator g of $g(p)$ is selected this generator must also be shared between both parties. The generator is a primitive root modulo of p. A number g is a **primitive root modulo of** p if:

$$\sum_{k=1}^{p-1} \left(g^k \pmod{p}\right) = \sum_{n=1}^{p-1} n = \frac{(p-1)p}{2}$$

This means that the sum of the generator raised to the power of *1-p modulus p* will always be the same as the sum of all integers from *1* to *1-p*. Therefor the expression can also be written as:

$$\{g^1 \pmod p),\ g^2 \pmod p),\ g^3 \pmod p),\ \ldots,\ g^{p-1} \pmod p)\} = \{1, 2, 3, \ldots, p-1\}.$$

After the generator has been selected each client generates a secret number. Each party calculates a public key *pubk* that is to be shared using their *secret*.

$$pubk = \{g^{secret} \pmod{p}\}$$

The shared secret s can then be calculated by using the other parties public key, the clients secret and the large common prime p.

$$s = \{pubk^{secret} \pmod{p} \pmod{p}\}$$

Now assuming that the bad actor as can be seen in *Figure 12* has intercepted all traffic it will be to mathematically complex for the bad actor to calculate the shared secret based just on the large common prime p. Assuming the bad actor is trying the General Number Field Sieve (GNFS) we can calculate the computation approximate complexity using the following big O expression where n is the current complexity of p. If $p = 127$ then $n = 2^8$

$$O(n) = \exp((\log n)2/3).$$

Assuming that p is a 1024-bit number there is then the computation complexity of solving the shared secret will be

$$O(2^{1024}) = \exp((\log 2^{1024})2/3).$$

2^{1024} total combinations of p. According to the prime number theorem the amount of primes will be the natural logarithm of the complexity of p therefore there are approximately *709.63* potential valid numbers.

$$\ln(2^{1024}) = 1024 \cdot 0.693 \approx 709.63.$$

Writer's note: *Include a graph of the running computational complexity of a brute-force attack ie.* $2^1 ... 2^n$

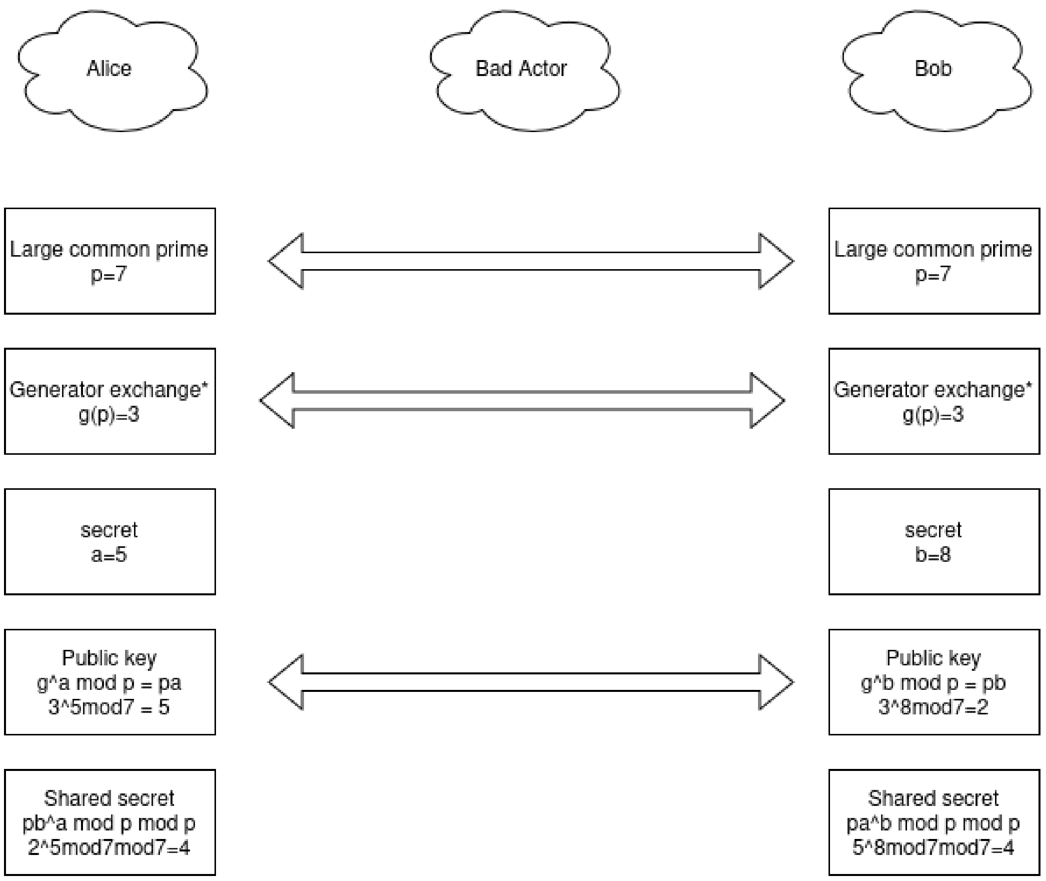


Figure 12: An illustration of the Diffie-Hellman key exchange protocol.
*The generator number is a primitive root modulo of p.

References

[1] *@lightside-Instruments/Red-Netconf.* http://flows.nodered.org/node/@lightside-instruments/red-netconf. (Visited on 04/21/2025).

[2] *Accenture — Let There Be Change.* https://www.accenture.com/no-en. (Visited on 04/27/2025).

[3] *ACM Digital Library.* https://dl.acm.org/. (Visited on 03/28/2025).

[4] *Agile Alliance.* https://www.agilealliance.org/. June 2015. (Visited on 03/28/2025).

[5] "Agile Software Development". In: *Wikipedia* (Apr. 2025). (Visited on 04/27/2025).

[6] Martin Björklund. *The YANG 1.1 Data Modeling Language.* Request for Comments RFC 7950. Internet Engineering Task Force, Aug. 2016. DOI: 10.17487/RFC7950. (Visited on 01/12/2025).

[7] Jeff Brehm. *AGILE TRANSFORMATION? FOR COMPLEX SYSTEMS? ...NO WAY!.* Oct. 2025. (Visited on 04/27/2025).

[8] "Continuous Delivery". In: *Wikipedia* (Jan. 2025). (Visited on 04/27/2025).

[9] *Deloitte — Audit, Consulting, Financial, Risk Management, Tax Services.* https://www.deloitte.com/global/en.html. (Visited on 04/27/2025).

[10] *DORA — Get Better at Getting Better.* https://dora.dev. (Visited on 04/27/2025).

[11] Rob Enns et al. *Network Configuration Protocol (NETCONF).* Request for Comments RFC 6241. Internet Engineering Task Force, June 2011. DOI: 10.17487/RFC6241. (Visited on 01/12/2025).

[12] *Extensible Markup Language (XML).* https://www.w3.org/XML/. (Visited on 05/07/2025).

[13] "Extreme Programming". In: *Wikipedia* (Apr. 2025). (Visited on 04/27/2025).

[14] *Framework.* https://framework.scaledagile.com/. (Visited on 04/27/2025).

[15] *Google Scholar*. https://scholar.google.com/. (Visited on 03/28/2025).

[16] *Grafana: The Open and Composable Observability Platform*. https://grafana.com/. (Visited on 01/20/2025).

[17] Joar Heimonen. *Slenderman00/Netconfaggregator*. May 2025. (Visited on 05/07/2025).

[18] *Home — Scrum.Org*. https://www.scrum.org/index. (Visited on 03/28/2025).

[19] *IEEE Xplore*. https://ieeexplore.ieee.org/Xplore/home.jsp. (Visited on 03/28/2025).

[20] "Kanban". In: *Wikipedia* (Apr. 2025). (Visited on 04/27/2025).

[21] "Kanban (Development)". In: *Wikipedia* (Mar. 2025). (Visited on 04/27/2025).

[22] *Lightside Instruments AS – YANG Model Network Managed Instruments*. (Visited on 01/20/2025).

[23] *Low-Code Programming for Event-Driven Applications : Node-RED*. https://nodered.org/. (Visited on 03/29/2025).

[24] *Manifesto for Agile Software Development*. http://agilemanifesto.org/. (Visited on 06/02/2024).

[25] *Node-Yuma123*. https://www.npmjs.com/package/node-yuma123. Apr. 2025. (Visited on 04/22/2025).

[26] *Node.Js — Run JavaScript Everywhere*. https://nodejs.org/en. (Visited on 03/29/2025).

[27] *Organisations-*. https://www.peoplecert.org/Organisations. (Visited on 03/28/2025).

[28] "Pair Programming". In: *Wikipedia* (Nov. 2024). (Visited on 04/27/2025).

[29] *Phased Project Planning Guidelines*. Aug. 1968. (Visited on 04/27/2025).

[30] *PRISMA Statement*. https://www.prisma-statement.org. (Visited on 03/28/2025).

[31] "Scrum (Rugby Union)". In: *Wikipedia* (Apr. 2025). (Visited on 04/27/2025).

[32] *Scrum Alliance - Find Courses for Scrum and Agile Certifications.* https://www.scrumalliance.org. (Visited on 03/28/2025).

[33] “The New New Product Development Game”. In: *Harvard Business Review* (). ISSN: 0017-8012. (Visited on 04/27/2025).

[34] Vladimir Vassilev. *Vlvassilev/Yuma123.* Mar. 2025. (Visited on 04/27/2025).

[35] Christiaan Verwijs and Daniel Russo. “A Theory of Scrum Team Effectiveness”. In: *ACM Transactions on Software Engineering and Methodology* 32.3 (July 2023), pp. 1–51. ISSN: 1049-331X, 1557-7392. DOI: 10.1145/3571849. (Visited on 04/27/2025).

[36] “Waterfall Model”. In: *Wikipedia* (Feb. 2025). (Visited on 04/27/2025).

[37] *XML Path Language (XPath) 3.1.* https://www.w3.org/TR/xpath-31/. (Visited on 05/07/2025).