Building new IoT Services Easily with Open Hardware and Lhings

José Antonio Lorenzo – Lyncos Technologies S. L.
José Pereda – Universidad de Valladolid

05th February 2014

Who the heck are these guys??!!

José Antonio Lorenzo

- PhD in Physics
- Lyncos Technologies, Spain
- JavaEE, embedded
- http://blog.lhings.com
- @joanlofe

Who the heck are these guys??!!

José Pereda

- PhD in Industrial Engineering
- University of Valladolid, Spain
- JavaFX, Embedded
- http://jperedadnr.blogspot.com
- @JPeredaDnr

lyncos Technologies

A startup based in Barcelona whose aim is to add value to products by connecting them to the Internet.

Lhings

Lhings, a cloud platform to provide connectivity through the Internet to all kind of devices, is the main product of Lyncos.

www.lhings.com

Tasty Recipes
==========

Recipe 12: Internet of things project

INGREDIENTS:

Some devices:
- 1 Arduino board
- 2 Raspberry Pi

Sensors:
- 1 thermometer
- 1 gas sensor
- 2 presence sensors

Actuators:
- 1 motor
- 3 relays

PREPARATION:

1. Design your circuits and build them.
2. Program your devices so they work as expected while taking care of low level programming issues.
3. Design your communications.
4. Program all your networking code.

8

- Security: mainly confidentiality and authentication
- Networking issues: firewalls, residential routers, push
- Scalability: how to manage hundreds of devices → message routing, connection management
- Reliability: quality of service, availability of the system

- Security: mainly confidentiality and authentication
- Networking issues: firewalls, residential routers, push
- Scalability: how to manage hundreds of devices → message routing, connection management
- Reliability: quality of service, availability of the system

A lot of effort in writing boilerplate code to build good connectivity, and not focusing on what your devices really should do

Lhings

```
public App()
{
    // Set initial behavior based on connectivity
    SetDataConnectivityOptions();
    // Set up handler for network status change
    NetworkInformation.NetworkStatusChanged += NetworkInformation_NetworkStatusChanged;
}
void NetworkInformation_NetworkStatusChanged(object sender)
{
    // Set behavior based on connectivity status
    SetDataConnectivityOptions();
}
void SetDataConnectivityOptions()
{
    // Get connection profile for currently active connection
    ConnectionProfile profile = NetworkInformation.GetInternetConnectionProfile();
    if (profile != null)
    {
        NetworkConnectivityLevel connectivityLevel = profile.GetNetworkConnectivityLevel();
```

Lhings

```
public App()
{
    // Set initial behavior based on connectivity
    SetDataConnectivityOptions();
    // Set up handler for network status change
    NetworkInformation.NetworkStatusChanged += NetworkInformation_NetworkStatusChanged;
}
void NetworkInformation_NetworkStatusChanged(object sender)
{
    // Set behavior based on connectivity status
    SetDataConnectivityOptions();
}
void SetDataConnectivityOptions()
{
    // Get connection profile for currently active connection
    ConnectionProfile profile = NetworkInformation.GetInternetConnectionProfile();
    if (profile != null)
    {
        NetworkConnectivityLevel connectivityLevel = profile.GetNetworkConnecti
```

```
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>

void error(const char *msg)
{
    perror(msg);
    exit(0);
}

int main(int argc, char *argv[])
{
    int sockfd, portno, n;
    struct sockaddr_in serv_addr;
    struct hostent *server;
    char buffer[256];
    if (argc < 3) {
        fprintf(stderr,"usage %s hostname port\n", argv[0]);
        exit(0);
    }
    portno = atoi(argv[2]);
    sockfd = socket(AF_INET, SOCK_STREAM, 0);
    if (sockfd < 0)
        error("ERROR opening socket");
```

Jfokus 2014, Stockholm

Lhings

```
public App()
{
    // Set initial behavior based on connectivity
    SetDataConnectivityOptions();
    // Set up handler for network status change
    NetworkInformation.NetworkStatusChanged += NetworkInformation_NetworkStatusChanged;
}
void NetworkInformation_NetworkStatusChanged(object sender)
{
    // Set behavior based on connectivity status
    SetDataConnectivityOptions();
}
void SetDataConnectivityOptions()
{
    // Get connection profile for currently active connection
    ConnectionProfile profile = NetworkInformation.GetInternetConnectionProfile();
    if (profile != null)
    {
        NetworkConnectivityLevel connectivityLevel = profile.GetNetworkConnecti
```

```
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>

void error(const char *msg)
{
    perror(msg);
    exit(0);
}

int main(int argc, char *argv[])
{
    int sockfd, portno
    struct sockad
    struct ho
                                ddr;
                    (0);
                    stderr,"usage %s hostname port\n", argv[0]);
    portno = atoi(argv[2]);
    sockfd = socket(AF_INET, SOCK_STREAM, 0);
    if (sockfd < 0)
        error("ERROR opening socket");
```

```
URLConnection connection = new URL(url + "?" + query).openConnection();
connection.setRequestProperty("Accept-Charset", charset);
InputStream response = connection.getInputStream();
```

Jfokus 2014, Stockholm

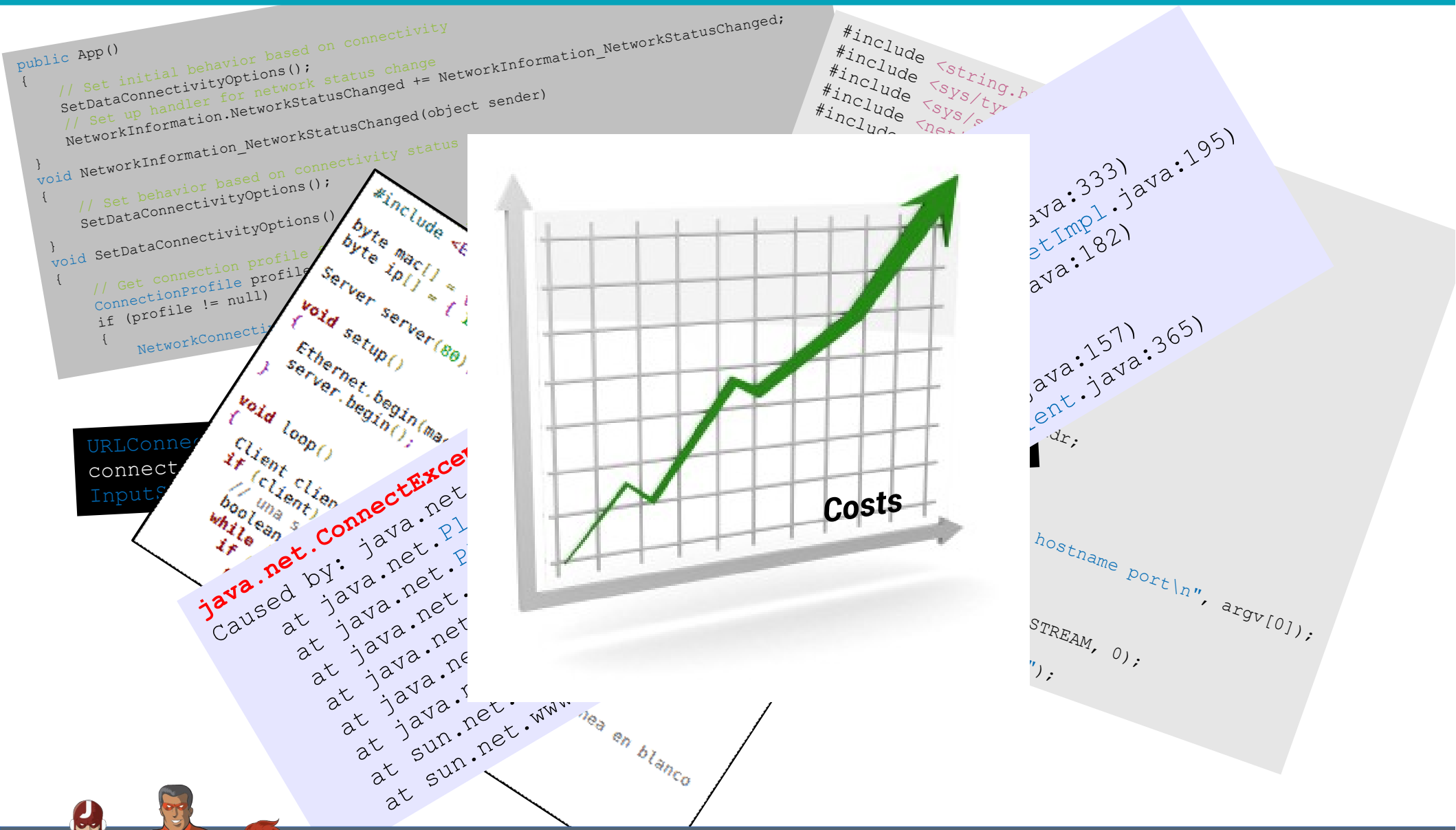
Lhings

Jfokus 2014, Stockholm

Lhings

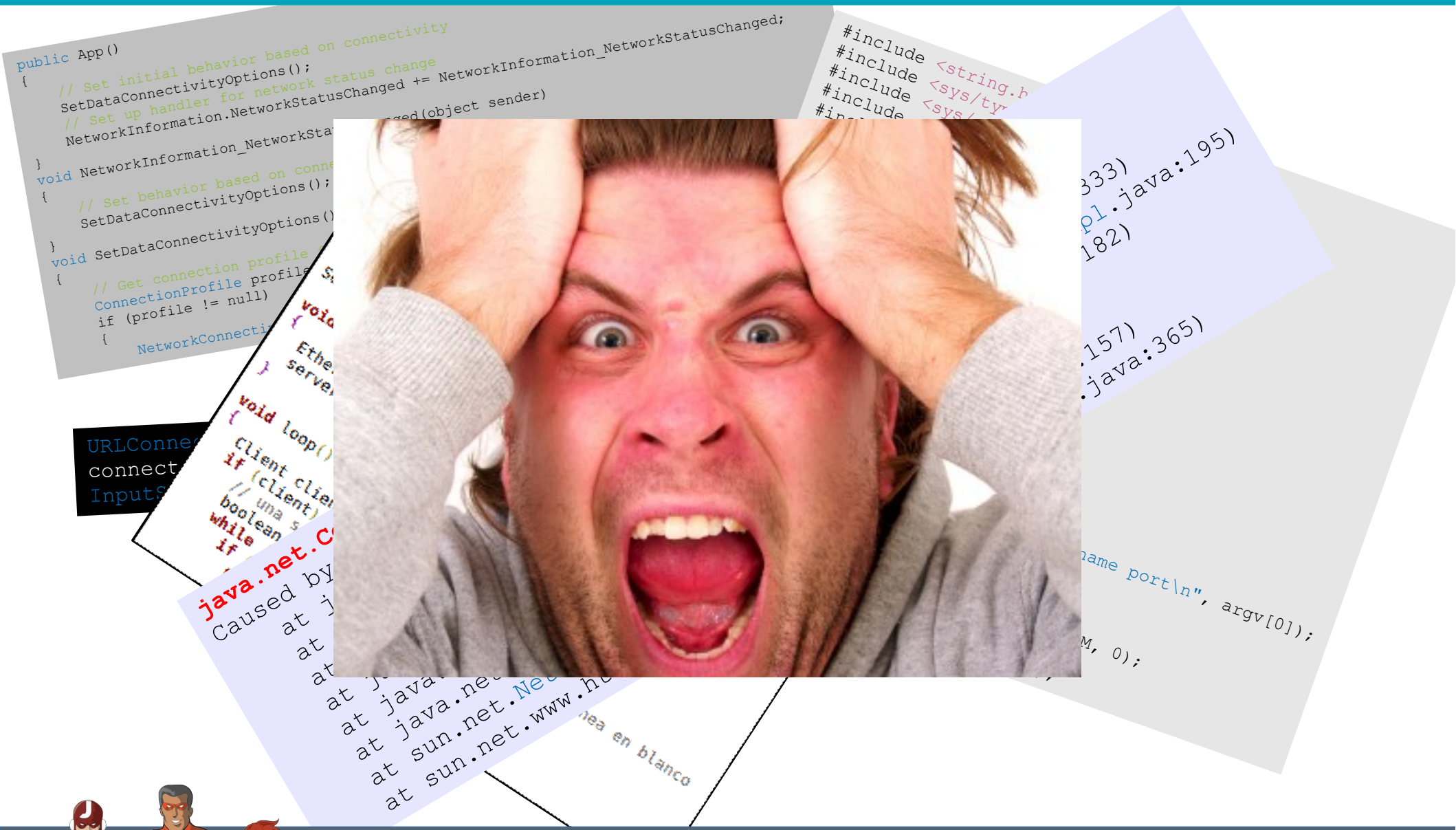
Jfokus 2014, Stockholm

Lhings



Jfokus 2014, Stockholm

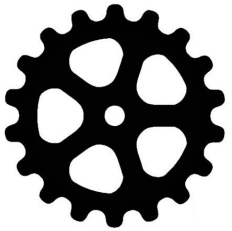
Lhings



Jfokus 2014, Stockholm

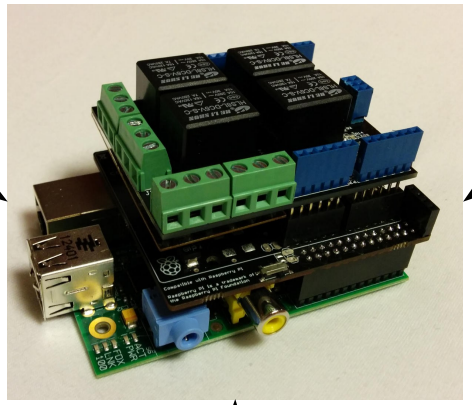
Lhings is a cloud service that allows you to connect, manage and control your devices easily without worrying about the nitty gritty details of networking

The three main concepts behind Lhings...



Actions

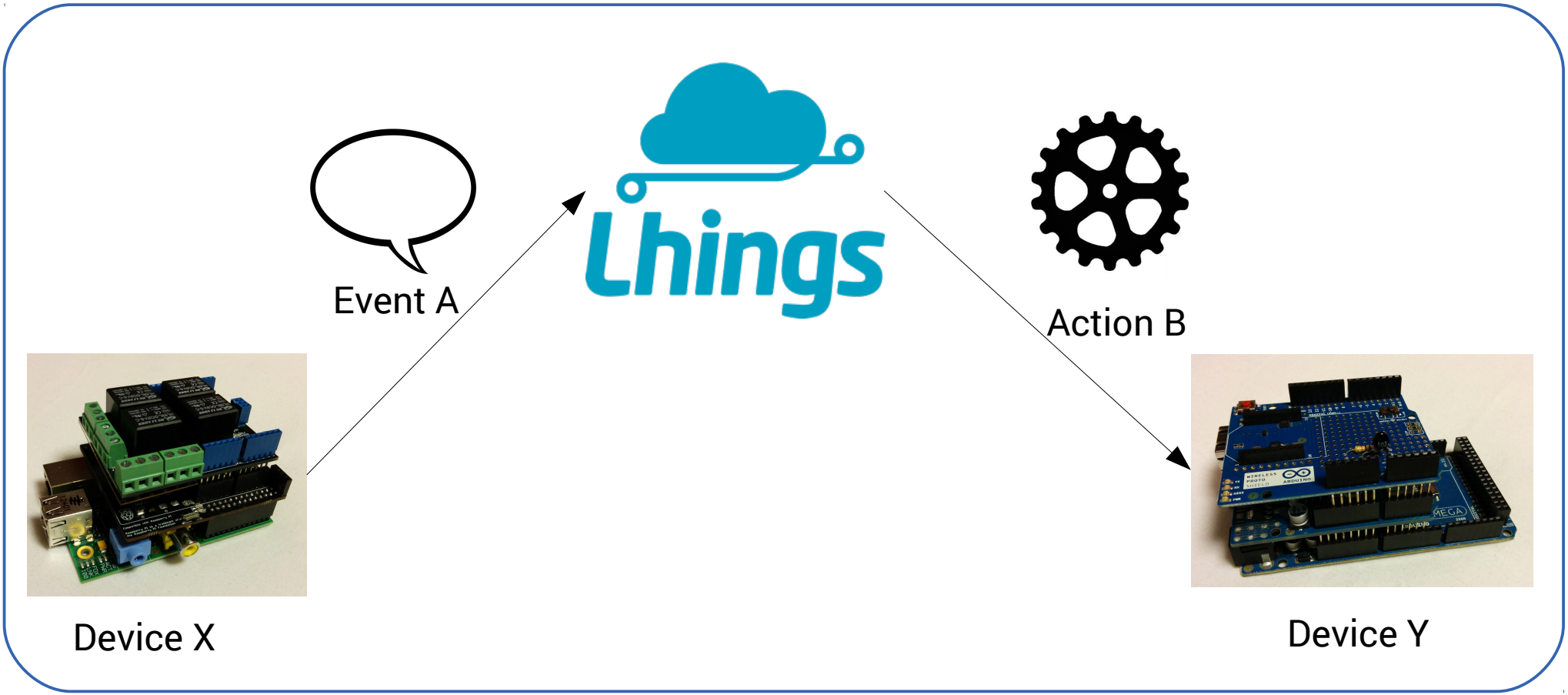
Device



Events

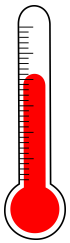
Status

… and the fourth: the rule concept

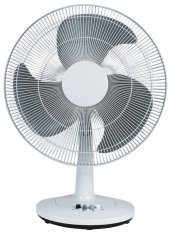


Internet

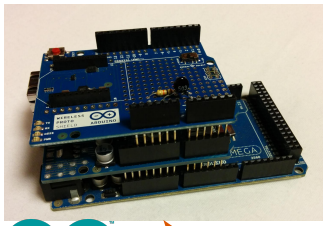
Visual temperature feedback



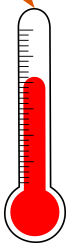
Temperature sensor



Fan



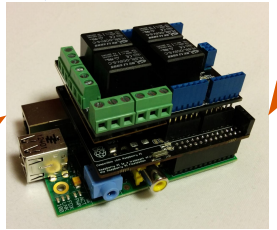
ARDUINO

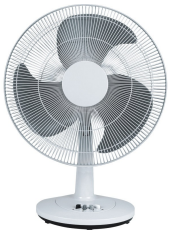


Temperature sensor

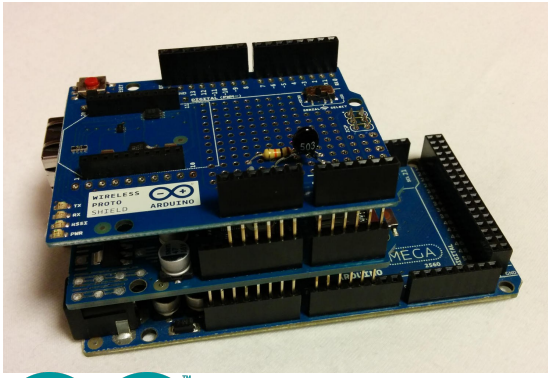
Lhings

Visual temperature feedback





Fan



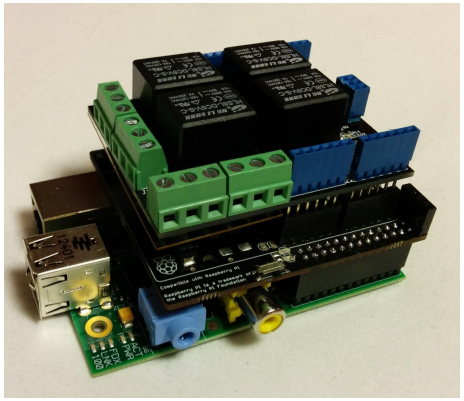
ARDUINO

Arduino board with temperature sensor:

- **Events**:
  - Temperature
  - Above threshold
  - Below threshold

- **Status**:
  - Temperature.

Raspberry Pi with relay shield:



- **Actions**:
  - Hue on
  - Hue off
  - Set color
  - Fan on
  - Fan off

- **Status**:
  - Fan status

And now let's go with the live demo!

What we get?

- We forgot completely about networking code.
- All messages transmitted are secure.
- We don't need to worry about firewalls, no need to open ports.
- Ability to push is provided out of the box.
- You get a web control panel for your devices automatically.
- Easy to scale.
- Reliability.

Do you want to try it? Our gift to you for coming to this talk is a free invitation to test our beta version.

The Lhings libraries and the code for this demo is available on Github:

https://github.com/lhings