
THE THINGS
NETWORK

RFC: Network Architecture

October 2015

Johan Stokking

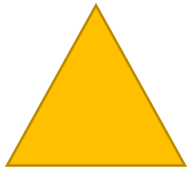
Tech Lead, The Things Network

Timeline

- July 2015: Validate basic connectivity and routing by building a prototype
- August 2015: Demonstrated prototype during The Things Network Conference
- September 2015: Gather input from experienced community members
- October 2015: Present RFC and validate design with experts
- November 2015 onwards: Implement and test components

Design Principles

- Fully compliant with the LoRaWAN 1.x specification
- Decentralized architecture: no single point of control
- Support for geographical segmentation to keep the data close
- Efficient routing: reduce bandwidth, computations and packet drop rate
- Trust-based model for both the gateway owners and application developers
- End-to-end encryption: application keys remain secret to the application



Node

Any LoRaWAN 1.x compliant device

THE THINGS NETWORK

LoRaWAN and The Things Network

The Things Network is fully compliant with the LoRaWAN 1.x specification, including secure communication, over-the-air-activation, adaptive data rates and bi-directional data transmission. Go to http://www.lora-alliance.org for more information about LoRaWAN.

Because of the limited device address space in LoRaWAN, The Things Network uses segmentation of gateways. Segments with the size of states or countries support overlap and handing over moving nodes to neighboring segments.

LoRaWAN provides builtin AES encryption using 128 bit keys on both the network and the application level. Application keys remain secret to the application to enable end-to-end encryption. The Things Network back-end components provide mitigation of various man-in-the-middle attacks, for example by checking the message integrity and frame counters.

THE THINGS NETWORK

Core Components

Gateway

Send data to and receive data from nodes

Router

Routes raw packets from gateways to brokers

Broker

Decoupling from router to handler

Network Server

Node data rate and frequency management

Handler

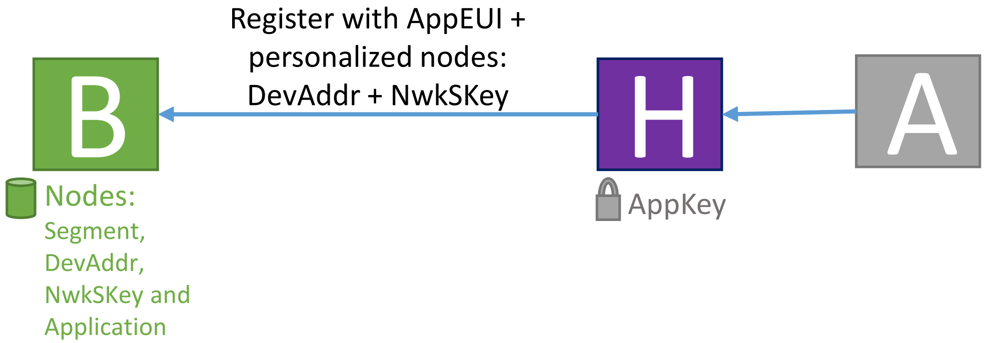
Decryption, deduping, works on behalf of apps

Application

Application or IoT cloud platform

Application Registration

Application Registration

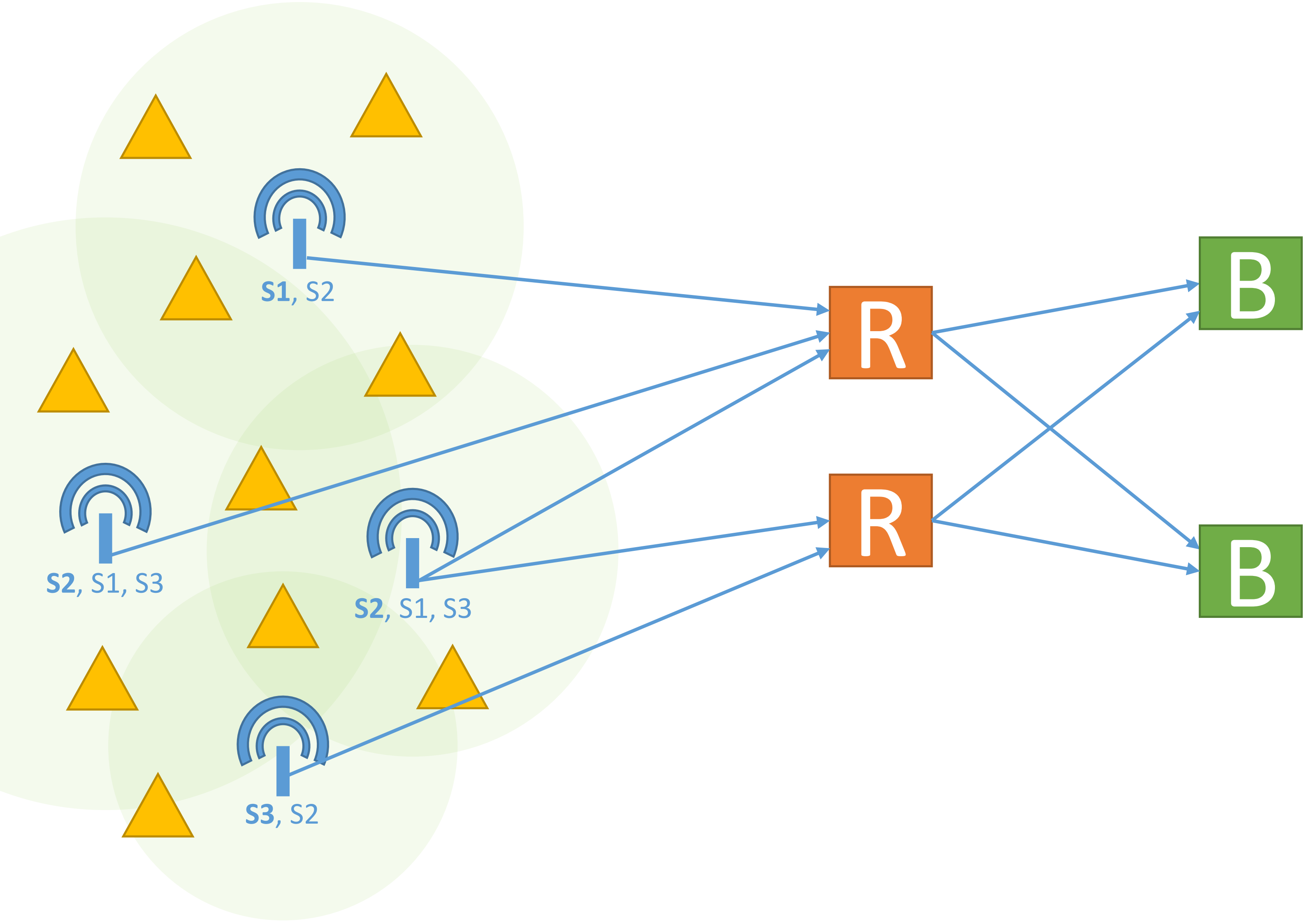


The Handler acts on behalf of the Application and informs the trusted Broker that it accepts activation for its globally unique AppEUI and it optionally provides a list of personalized devices.

The Broker persists the segments of the nodes, their device addresses and network session keys. Geographical segmentation enables scaling and reusing devices addresses. The network security keys are used to check if the message belongs indeed to an application by checking the message integrity (MIC).

Gateway Setup

Gateway Setup



The Things Network uses geographical segments in which gateways are grouped. Gateways have a primary segment ID and optionally a secondary or tertiary ID of the segments that they are close by to. The segments are used to extend the limited address space for devices when using OTAA.

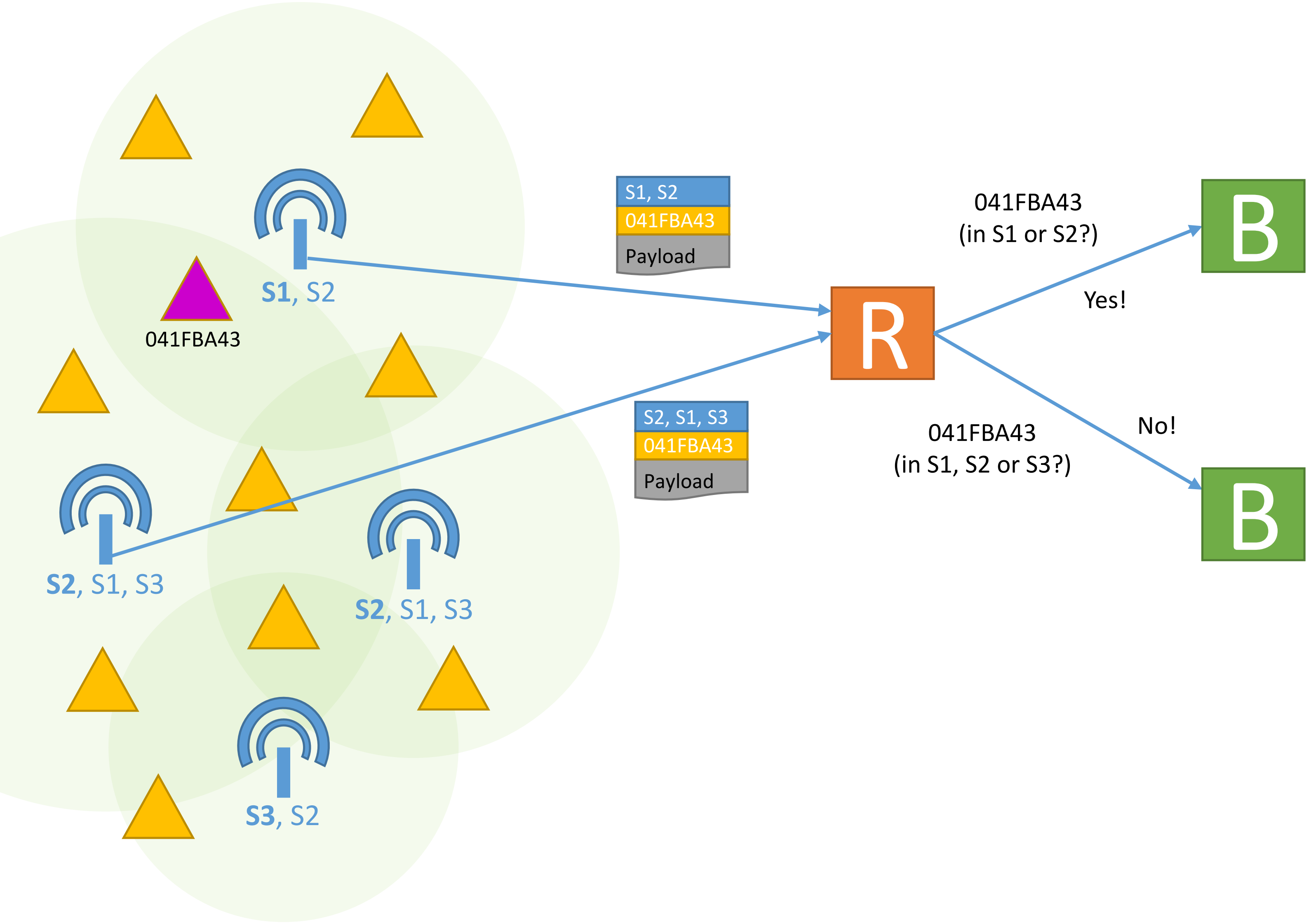
Initially, Routers are preconfigured to connect to Brokers. Later, a discovery mechanism will be introduced, for example a peer-to-peer-like tracking or a blockchain.

The segments S1, S2 and S3 shown here are illustrative only. In reality, the segments will have the size of a state or a country, depending on the number of devices in the area.

THE THINGS NETWORK

Packet Routing

Packet Routing

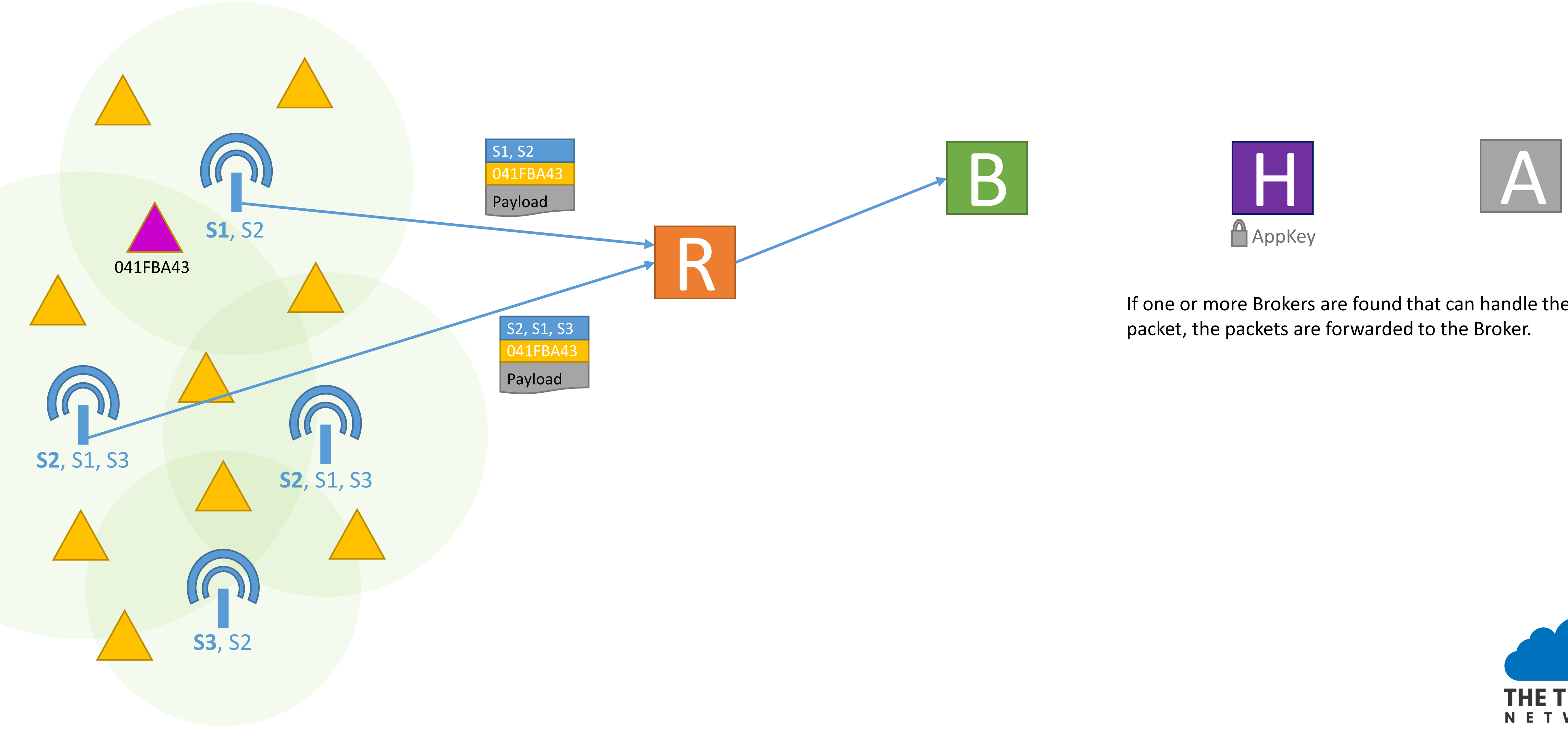


Packets sent from nodes are forwarded to the Routers that are configured on the gateway. In this example, packets from node with address 041FBA43 are received by two gateways. Both forward the packet to the Router.

The Router checks its local cache to see if it already knows a Broker than can handle the packet from this node in the provided segment(s). If it doesn't, it asks the Brokers whether they can and updates its local cache.

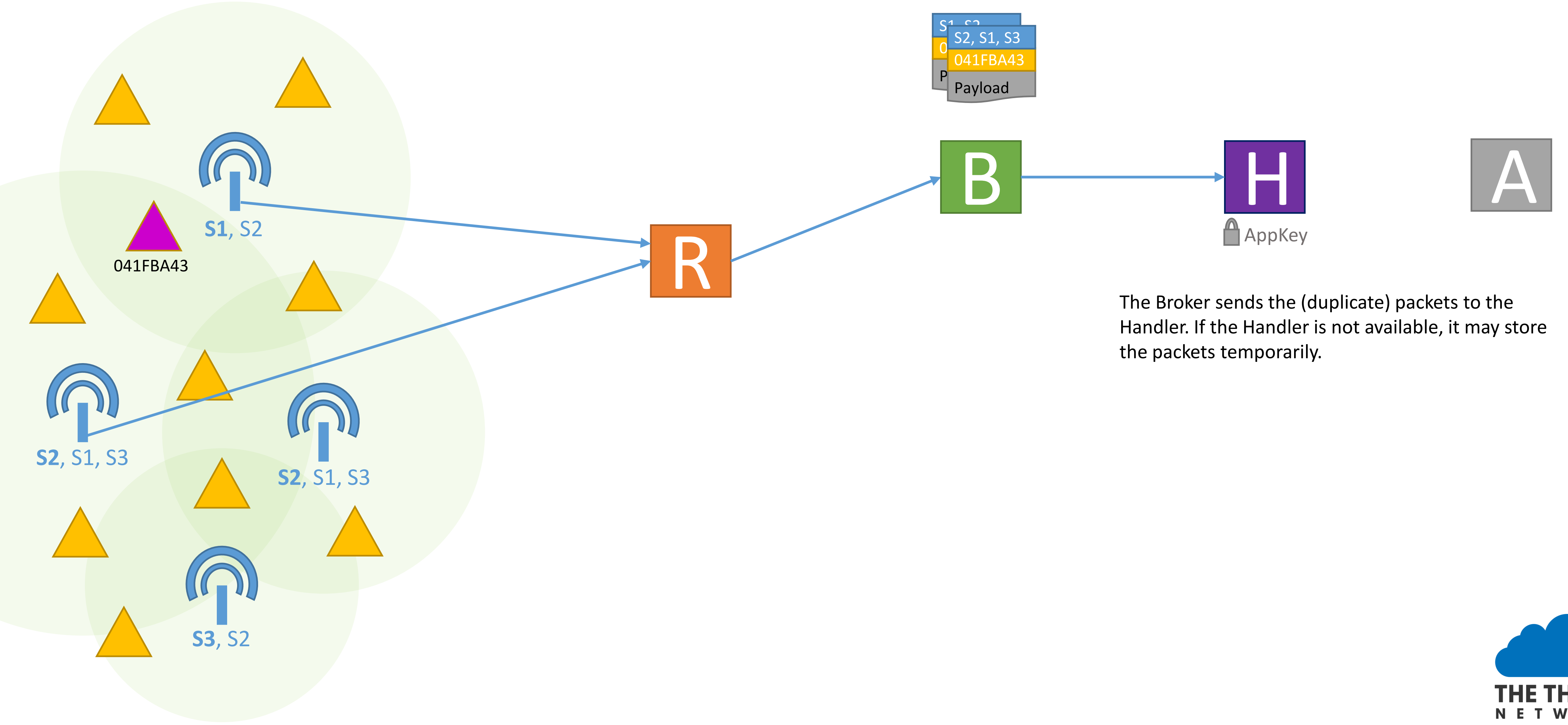
THE THINGS NETWORK

Packet Routing



If one or more Brokers are found that can handle the packet, the packets are forwarded to the Broker.

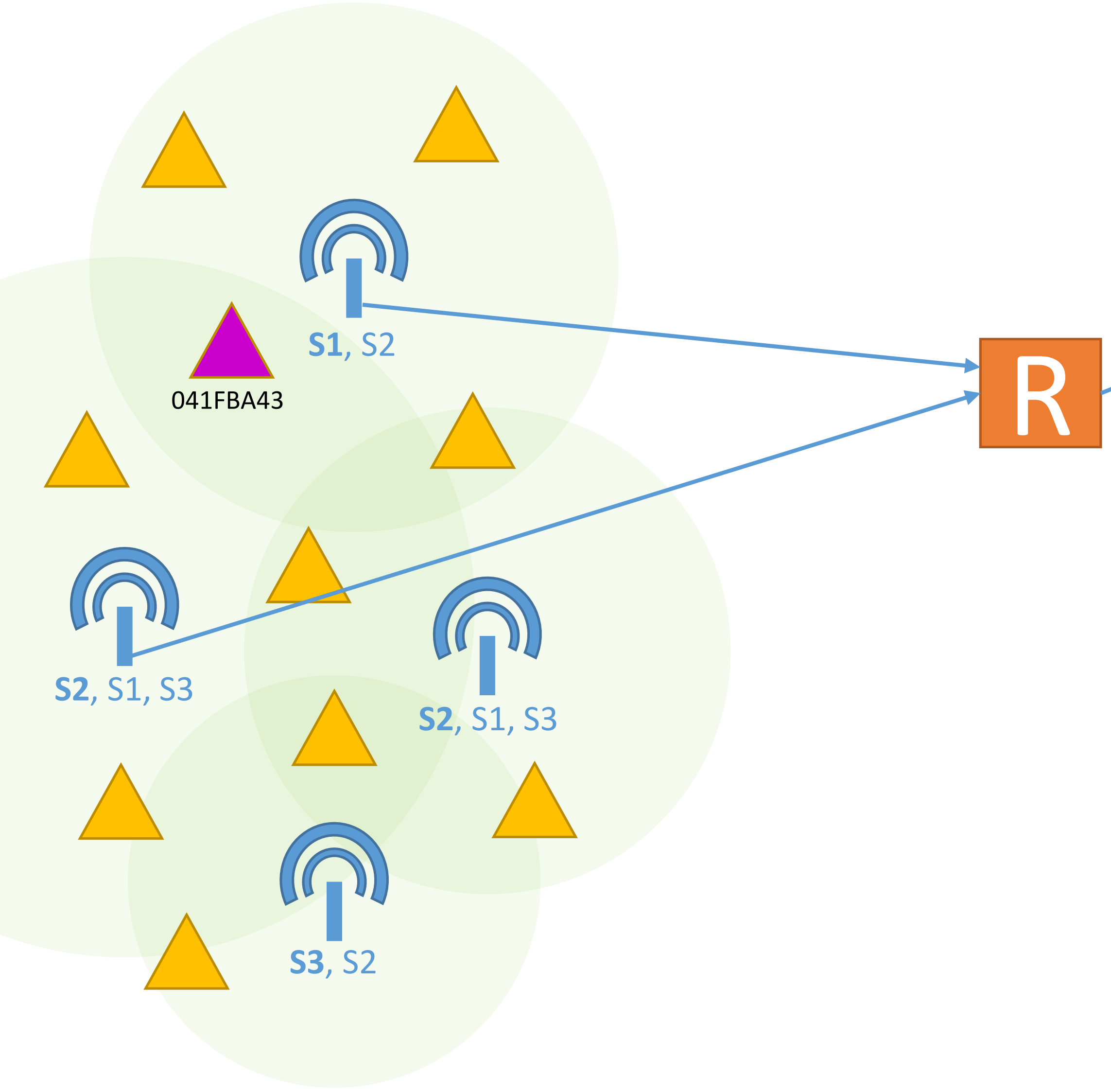
Packet Routing

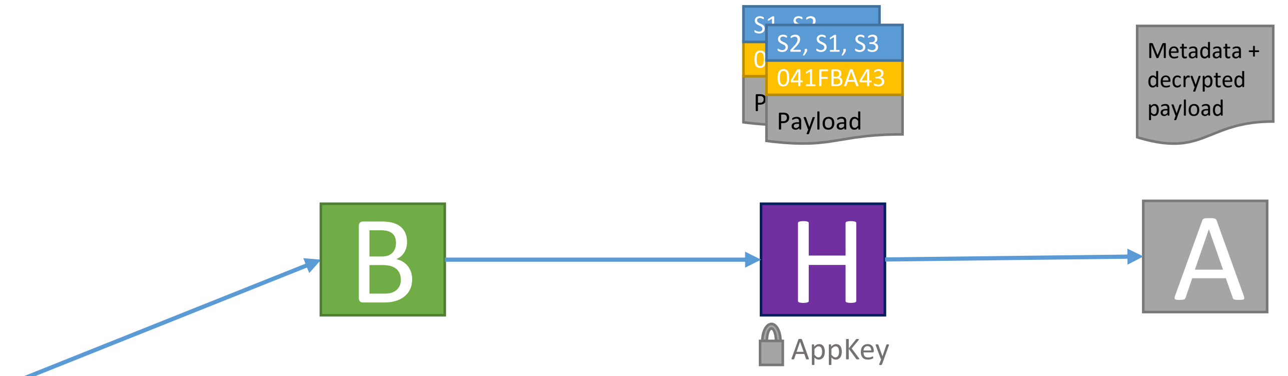


The Broker sends the (duplicate) packets to the Handler. If the Handler is not available, it may store the packets temporarily.

THE THINGS NETWORK

Packet Routing





The Handler decrypts and deduplicates the packets. Also, based on three or more duplicates, geolocalization can be performed using triangulation.

Over The Air Activation

THE THINGS
NETWORK

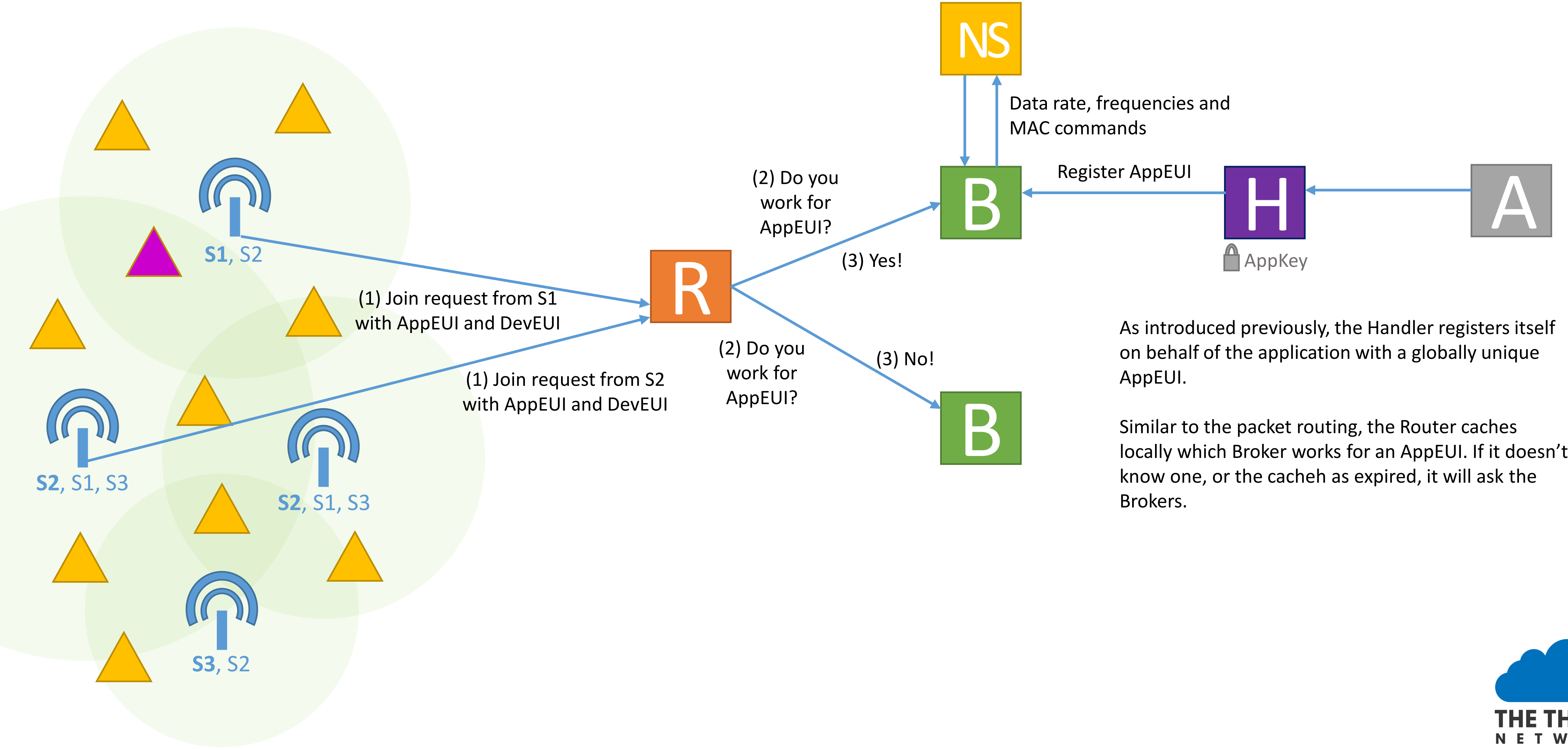
Over The Air Activation

LoRaWAN supports two types of registering nodes for an application: personalization and over the air activation (OTAA)

When using personalization, the application developer hardcodes the device address and network security key in the device.

When using OTAA, the device joins the network by providing a unique application identifier (AppEUI) and device identifier (DevEUI). The network negotiates with the application and generates a temporary device address and session security keys.

Over The Air Activation

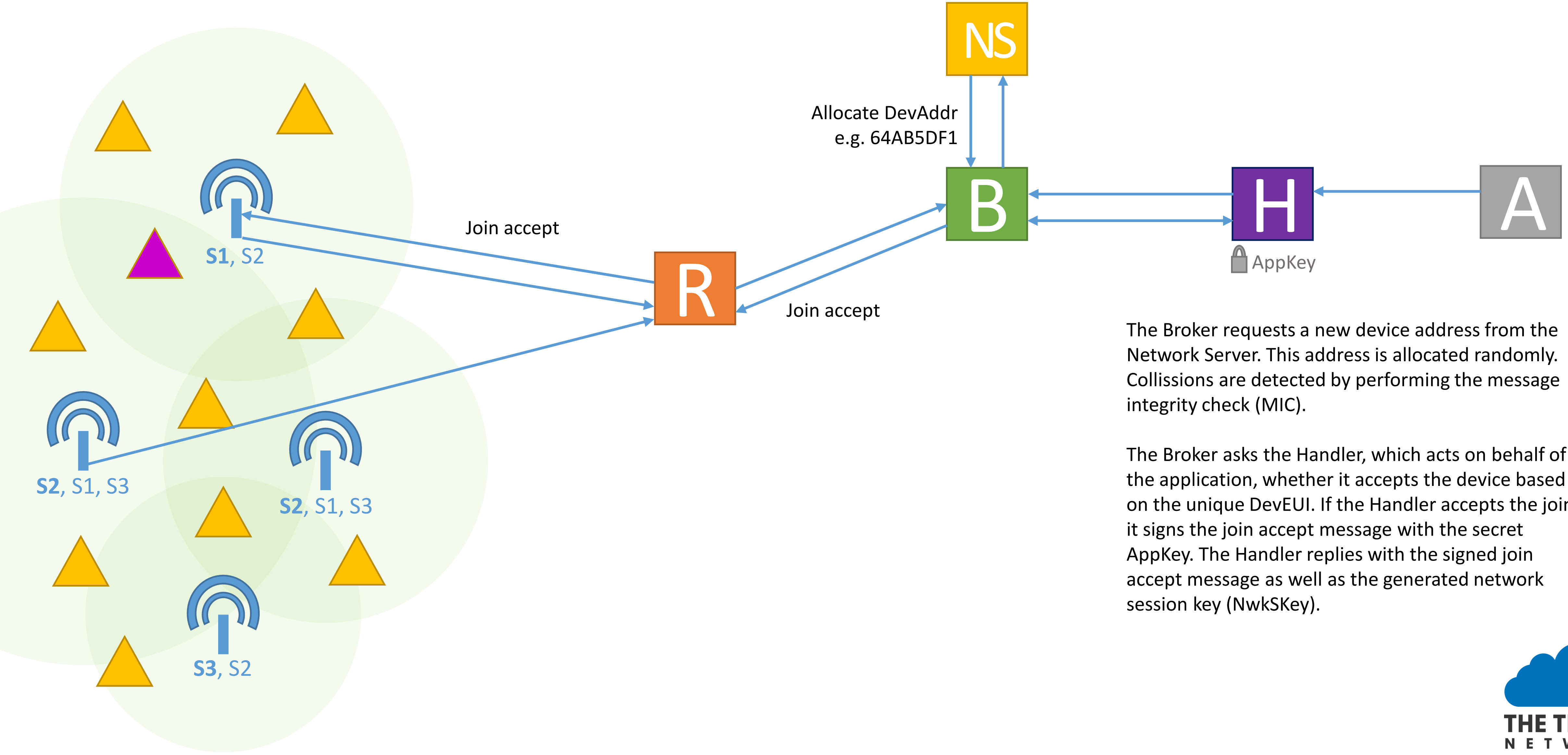


As introduced previously, the Handler registers itself on behalf of the application with a globally unique AppEUI.

Similar to the packet routing, the Router caches locally which Broker works for an AppEUI. If it doesn’t know one, or the cacheh as expired, it will ask the Brokers.

THE THINGS NETWORK

Over The Air Activation



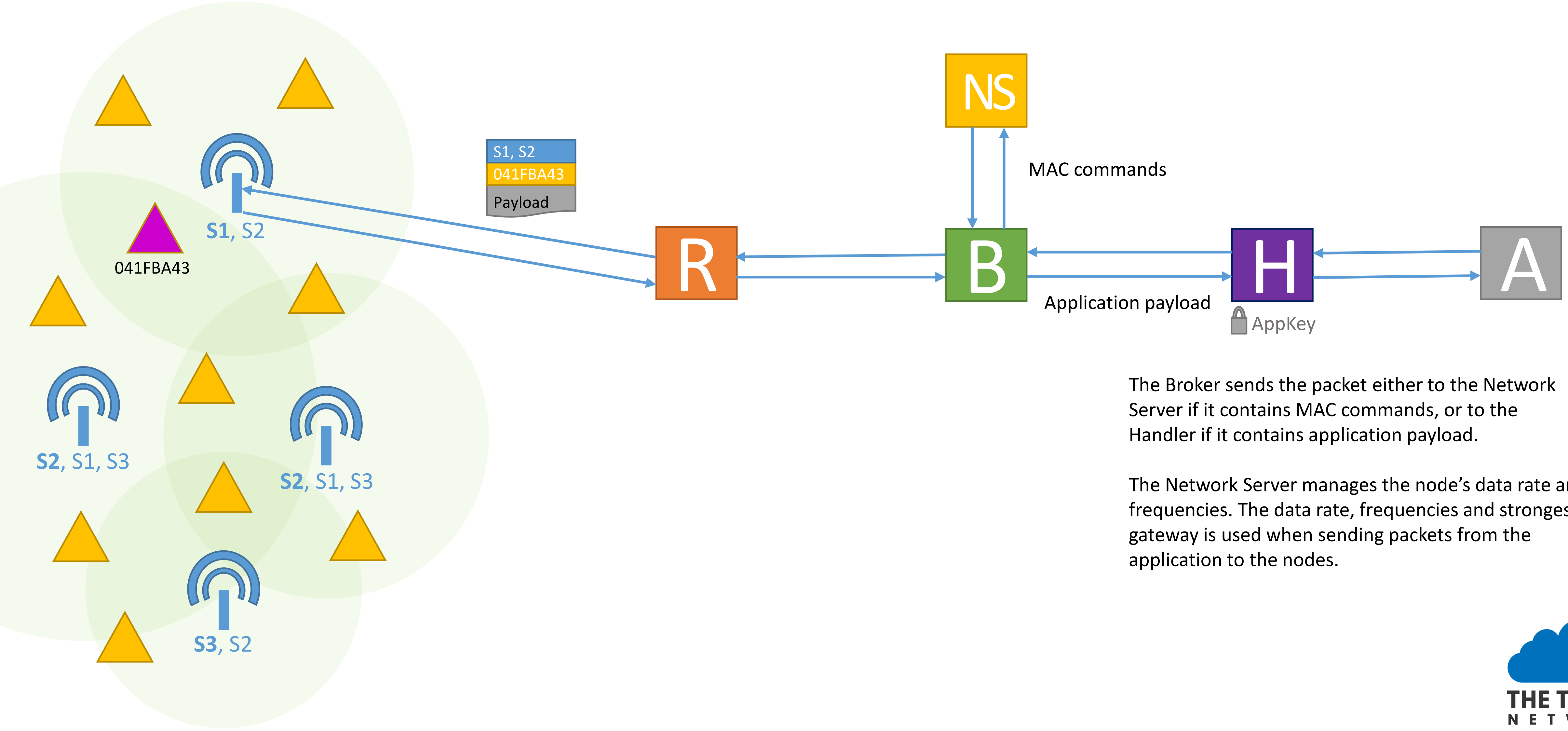
The Broker requests a new device address from the Network Server. This address is allocated randomly. Collissions are detected by performing the message integrity check (MIC).

The Broker asks the Handler, which acts on behalf of the application, whether it accepts the device based on the unique DevEUI. If the Handler accepts the join, it signs the join accept message with the secret AppKey. The Handler replies with the signed join accept message as well as the generated network session key (NwkSKey).

THE THINGS NETWORK

Network Management

Network Management



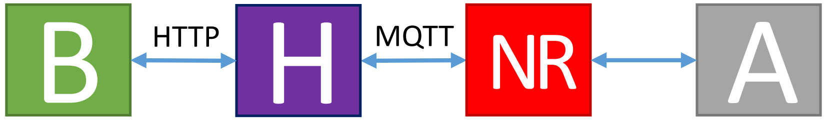
The Broker sends the packet either to the Network Server if it contains MAC commands, or to the Handler if it contains application payload.

The Network Server manages the node's data rate and frequencies. The data rate, frequencies and strongest gateway is used when sending packets from the application to the nodes.

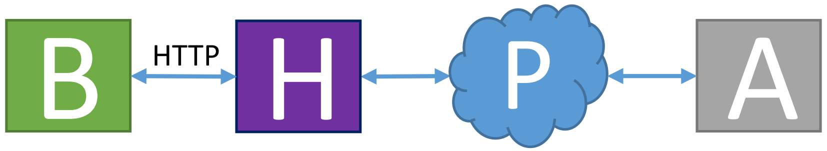
THE THINGS NETWORK

Application and Cloud Platform Integrations

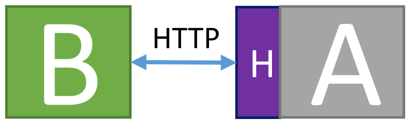
Connect Application with the open source The Things Handler



Connect Application with Node RED



Connect Application with AwS IoT, IBM Bluemix, FIWARE, etc



Integrate handler functionality in application

The Broker and the Handler are offered hosted services by the Foundation, but both services can be installed on-premise as well

THE THINGS NETWORK

Next Steps

- Present RFC and gather comments from experts
- Weekly tech update on Forum and mailinglist
- Expand the development teams

Teams

1. Architecture team draws on whiteboards
   - Overview of different components
   - Design for openness and decentralization
   - LoRaWAN compliance
2. Network access team develops in C/C++
   - Nodes
   - Gateway firmwares
3. Core team develops in Golang
   - Router, Broker, Network Server and Handler
4. Integration team makes things work together
   - Node RED setup and examples
   - AwS IoT, IBM Bluemix, FIWARE, IFTTT, Parse.com, etc
   - Bundling and containerizing
   - Code examples and libraries

How to Join a Team

Do you want to participate in the Architecture, Network Access, Core and/or Integration team?

Great! You will join the active The Things Network community.

- Register on our forum: http://forum.thethingsnetwork.org
- Sign up for the newsletter: http://thethingsnetwork.org (go to Join Team)
- View the issues on GitHub: http://github.com/TheThingsNetwork
- Send an e-mail: johan@thethingsnetwork.org

THE THINGS NETWORK


THE THINGS
NETWORK