APNIC 58

Informational presentation: Using IPv6 for Internet of Things

Guangliang PAN (Benny) / Wei WANG (Wesley)
APNIC 58 Open Policy Meeting
Wellington, New Zealand
6 September 2024

#apnic58

TE WHANGANUI A TARA
WELLINGTON,
AOTEAROA NEW ZEALAND
30 August – 6 September 2024

Original policy proposal

- **prop-161: Using IPv6 for Internet of Things (IoT)**

- IPv6 addresses can be allocated to Internet of Things for electronic smart devices and/or for hosting information of non-electronic items on the Internet.

Key requirement - Using IPv6 to host information of non-electronic items

- In some of the cases, the IoT industry needs to assign IPv6 to electronic smart devices as well as non-electronic items. The non-electronic items include company products and assets. IPv6 addresses will be used to host information of non-electronic items on the Internet for the purpose of identification, verification, and tracing.

Discussions on the policy mailing list

- There were active discussions on the policy mailing list regarding using IPv6 addresses for non-electronic items.
- Conclusion: “Using IPv6 addresses to host information of non-electronic items on the Internet” is acceptable in current policy. There is no need to change the policy.

Change policy proposal to informational presentation

- We will not seek consensus for policy proposal “prop-161 Using IPv6 for Internet of Things (IoT)”.

- This presentation is to share information with the APNIC community on practices of how to use IPv6 addresses to hosting information of non-electronic items.

Internet of everything by IPv6

- Assign single IPv6 address to each electronic smart device for direct point to point communications on the Internet.
- Map single IPv6 address with each non-electronic item for hosting unique information of that item on the Internet.
- It is a real Internet of everything.

IPv6 has a key benefit on anti-fake

- With RIR whois database information and secure routing, customers can trust the responding IPv6 address is belong to the factory who making the product.
- Due to huge numbers of IPv6 addresses, it is impossible to guess which IPv6 address mapped with which product.
- If a non-electronic item mapped with an IPv6 address which hosting unique information of that item on the Internet and can be checked any time, it will help stop fake products.

Using IPv6 address as IoT Identifier to host non-electronic item information

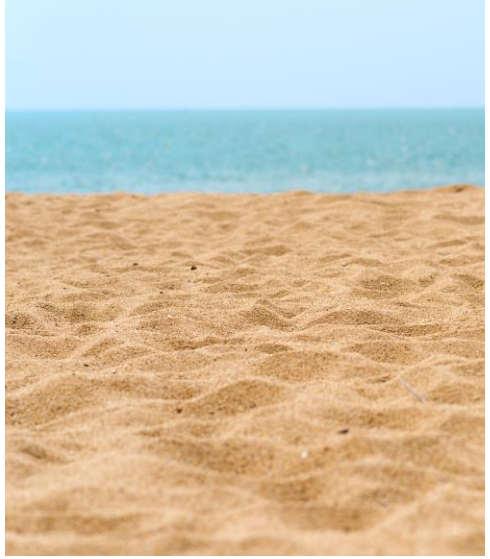
Wei WANG

wesleywangbeijing@gmail.com

#apnic58

TE WHANGANUI A TARA
WELLINGTON,
AOTEAROA NEW ZEALAND
30 August – 6 September 2024

IPv6 for Sand



“If the earth were made entirely out of 1 cubic millimetre grains of sand, then you could give a unique address to each grain in 300 million planets the size of the earth”

---- Wikipedia

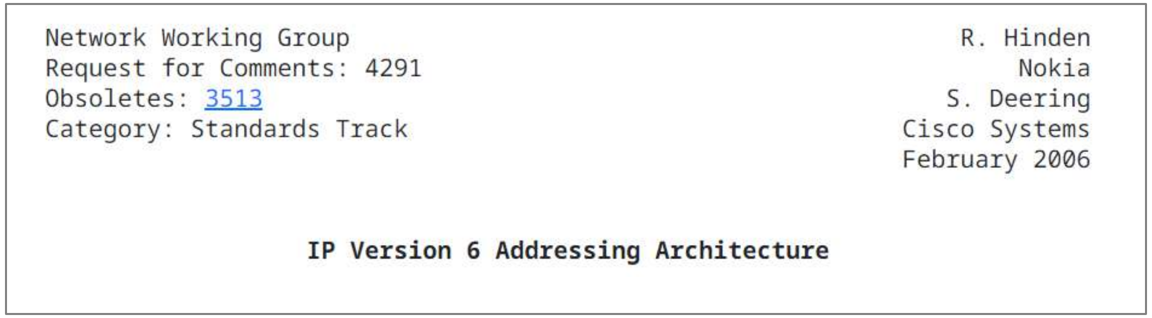
IPv6 for Sand

LARGE BAG
TARMAC
SHARP SAND
For use in creating concrete when mixed with cement & gravel

In practice, sand in deserts or beaches does not need and cannot be configured with an IPv6 address. 、

However, once sand is packaged or transformed into a product item, it may require an IPv6 address for identification and providindg access entry to the information associated with the item.

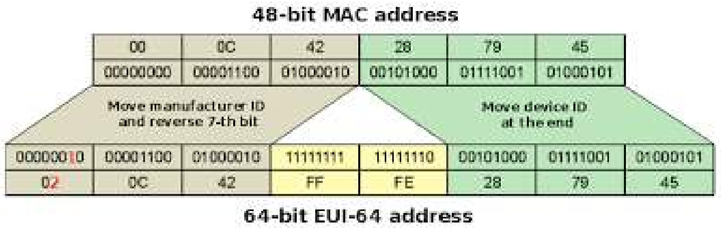
IPv6 address for everything



```
Network Working Group                                         R. Hinden
Request for Comments: 4291                                        Nokia
Obsoletes: 3513                                              S. Deering
Category: Standards Track                                 Cisco Systems
                                                          February 2006


                  IP Version 6 Addressing Architecture
```

"A single interface may also have multiple IPv6 addresses of any type(unicast, anycast, and multicast) or scope"
---- quoted form RFC 4291

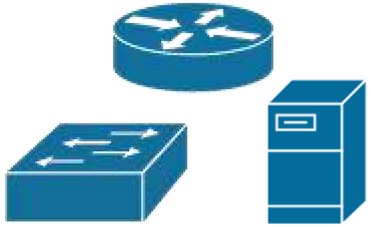


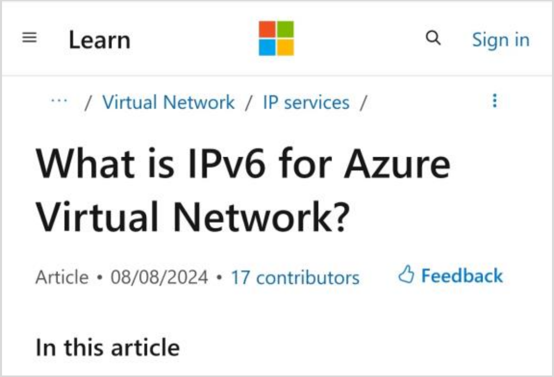
48-bit MAC address

00	0C	42	28	79	45
00000000	00001100	01000010	00101000	01111001	01000101

00000010	00001100	01000010	11111111	11111110	00101000	01111001	01000101
02	0C	42	FF	FE	28	79	45

64-bit EUI-64 address

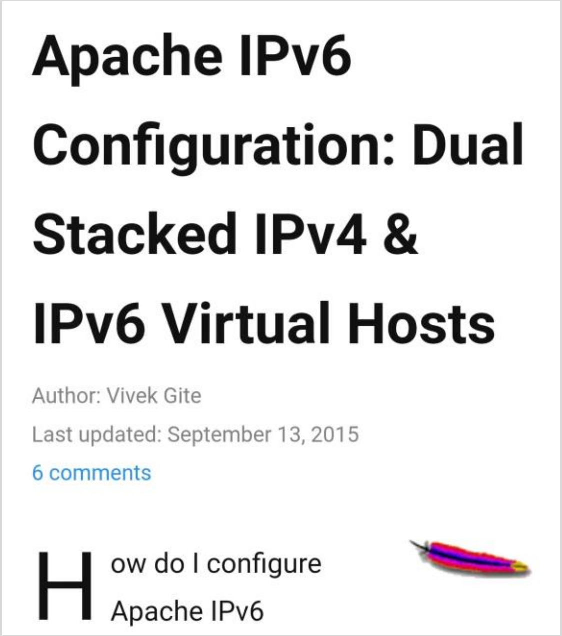




What is IPv6 for Azure Virtual Network?

Article • 08/08/2024 • 17 contributors

In this article



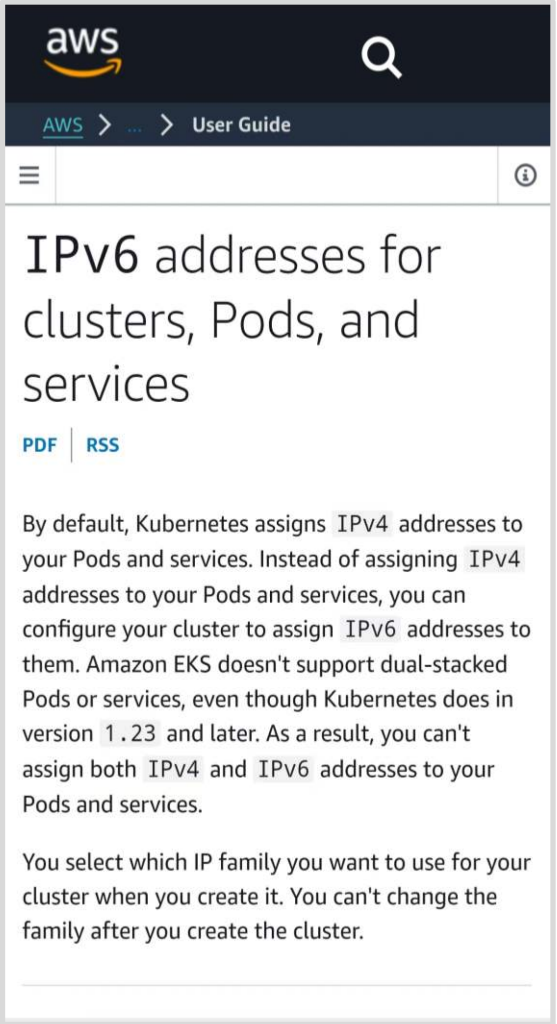
Apache IPv6 Configuration: Dual Stacked IPv4 & IPv6 Virtual Hosts

Author: Vivek Gite

Last updated: September 13, 2015

6 comments

How do I configure Apache IPv6

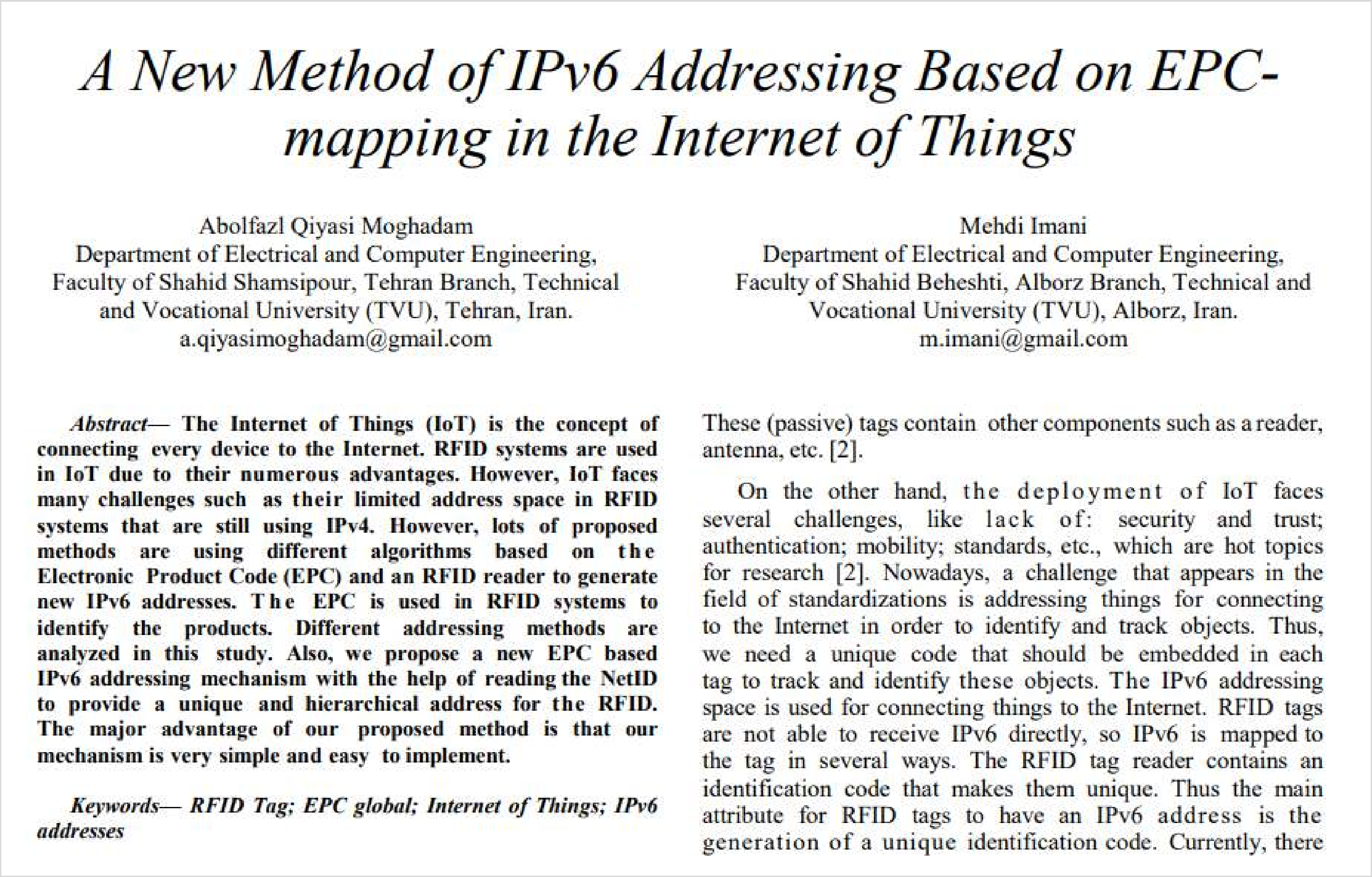


IPv6 addresses for clusters, Pods, and services

By default, Kubernetes assigns `IPv4` addresses to your Pods and services. Instead of assigning `IPv4` addresses to your Pods and services, you can configure your cluster to assign `IPv6` addresses to them. Amazon EKS doesn't support dual-stacked Pods or services, even though Kubernetes does in version `1.23` and later. As a result, you can't assign both `IPv4` and `IPv6` addresses to your Pods and services.

You select which IP family you want to use for your cluster when you create it. You can't change the family after you create the cluster.

IPv6 address for everything



A New Method of IPv6 Addressing Based on EPC-mapping in the Internet of Things

Abolfazl Qiyasi Moghadam
Department of Electrical and Computer Engineering,
Faculty of Shahid Shamsipour, Tehran Branch, Technical
and Vocational University (TVU), Tehran, Iran.
a.qiyasimoghadam@gmail.com

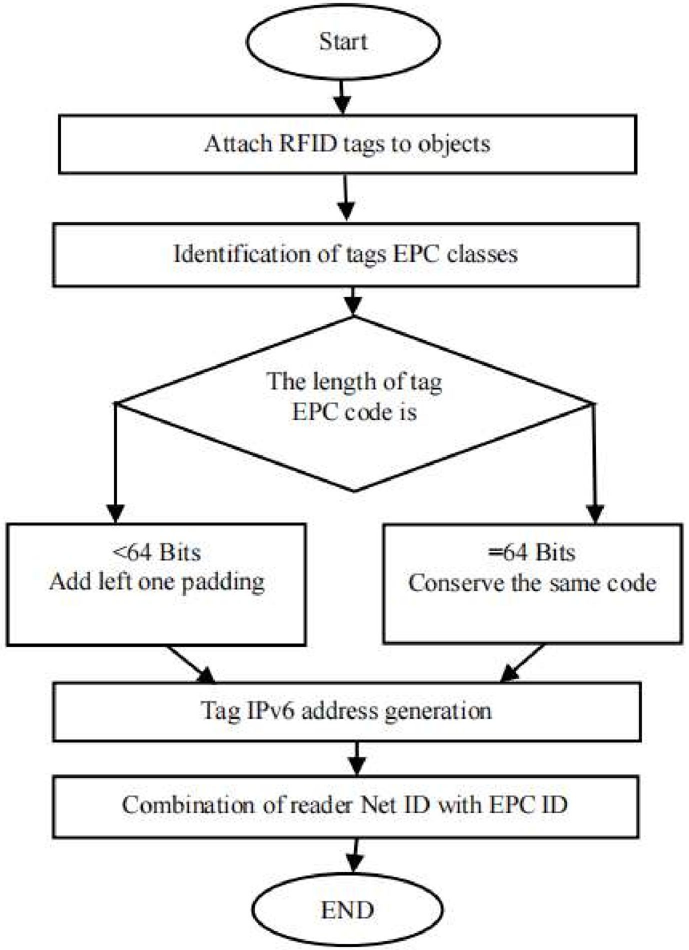
Mehdi Imani
Department of Electrical and Computer Engineering,
Faculty of Shahid Beheshti, Alborz Branch, Technical and
Vocational University (TVU), Alborz, Iran.
m.imani@gmail.com

***Abstract*— The Internet of Things (IoT) is the concept of connecting every device to the Internet. RFID systems are used in IoT due to their numerous advantages. However, IoT faces many challenges such as their limited address space in RFID systems that are still using IPv4. However, lots of proposed methods are using different algorithms based on the Electronic Product Code (EPC) and an RFID reader to generate new IPv6 addresses. The EPC is used in RFID systems to identify the products. Different addressing methods are analyzed in this study. Also, we propose a new EPC based IPv6 addressing mechanism with the help of reading the NetID to provide a unique and hierarchical address for the RFID. The major advantage of our proposed method is that our mechanism is very simple and easy to implement.**

***Keywords*— RFID Tag; EPC global; Internet of Things; IPv6 addresses**

These (passive) tags contain other components such as a reader, antenna, etc. [2].

On the other hand, the deployment of IoT faces several challenges, like lack of: security and trust; authentication; mobility; standards, etc., which are hot topics for research [2]. Nowadays, a challenge that appears in the field of standardizations is addressing things for connecting to the Internet in order to identify and track objects. Thus, we need a unique code that should be embedded in each tag to track and identify these objects. The IPv6 addressing space is used for connecting things to the Internet. RFID tags are not able to receive IPv6 directly, so IPv6 is mapped to the tag in several ways. The RFID tag reader contains an identification code that makes them unique. Thus the main attribute for RFID tags to have an IPv6 address is the generation of a unique identification code. Currently, there



Start

Attach RFID tags to objects

Identification of tags EPC classes

The length of tag EPC code is

<64 Bits
Add left one padding

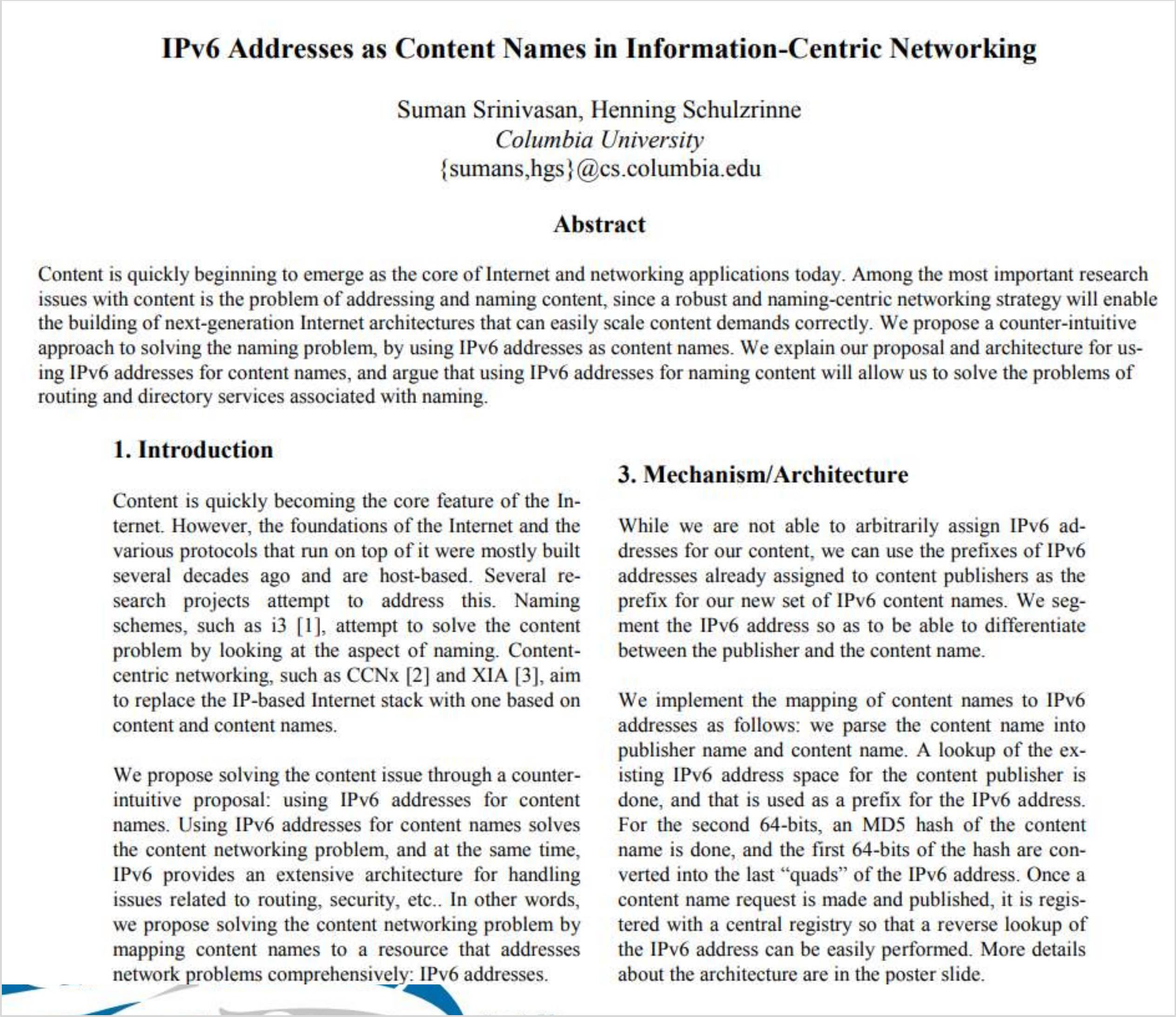
=64 Bits
Conserve the same code

Tag IPv6 address generation

Combination of reader Net ID with EPC ID

END

IPv6 address for everything



IPv6 Addresses as Content Names in Information-Centric Networking

Suman Srinivasan, Henning Schulzrinne
Columbia University
{sumans,hgs}@cs.columbia.edu

Abstract

Content is quickly beginning to emerge as the core of Internet and networking applications today. Among the most important research issues with content is the problem of addressing and naming content, since a robust and naming-centric networking strategy will enable the building of next-generation Internet architectures that can easily scale content demands correctly. We propose a counter-intuitive approach to solving the naming problem, by using IPv6 addresses as content names. We explain our proposal and architecture for using IPv6 addresses for content names, and argue that using IPv6 addresses for naming content will allow us to solve the problems of routing and directory services associated with naming.

1. Introduction

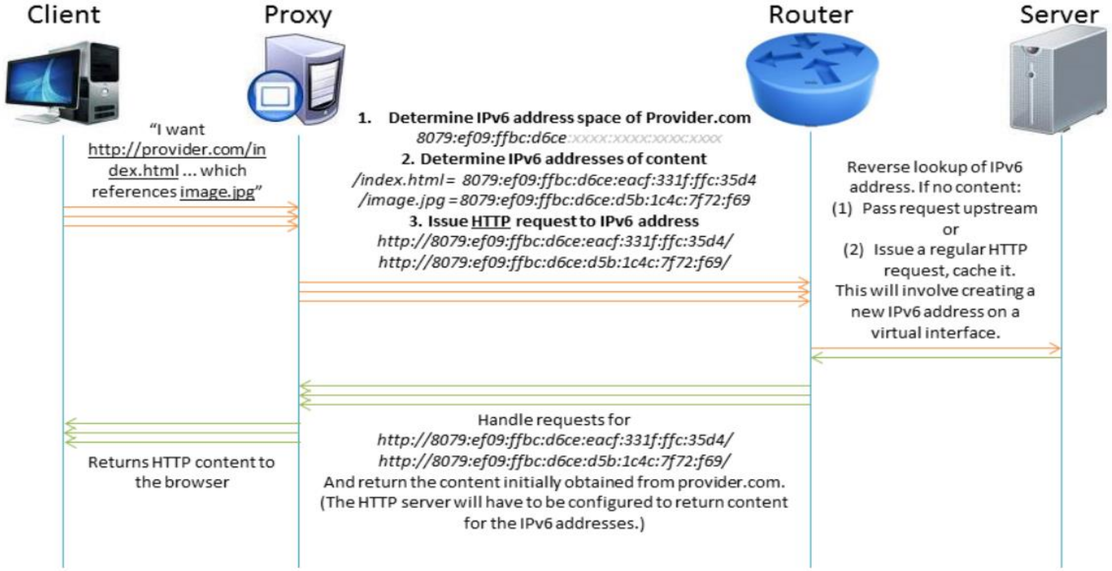
Content is quickly becoming the core feature of the Internet. However, the foundations of the Internet and the various protocols that run on top of it were mostly built several decades ago and are host-based. Several research projects attempt to address this. Naming schemes, such as i3 [1], attempt to solve the content problem by looking at the aspect of naming. Content-centric networking, such as CCNx [2] and XIA [3], aim to replace the IP-based Internet stack with one based on content and content names.

We propose solving the content issue through a counter-intuitive proposal: using IPv6 addresses for content names. Using IPv6 addresses for content names solves the content networking problem, and at the same time, IPv6 provides an extensive architecture for handling issues related to routing, security, etc.. In other words, we propose solving the content networking problem by mapping content names to a resource that addresses network problems comprehensively: IPv6 addresses.

3. Mechanism/Architecture

While we are not able to arbitrarily assign IPv6 addresses for our content, we can use the prefixes of IPv6 addresses already assigned to content publishers as the prefix for our new set of IPv6 content names. We segment the IPv6 address so as to be able to differentiate between the publisher and the content name.

We implement the mapping of content names to IPv6 addresses as follows: we parse the content name into publisher name and content name. A lookup of the existing IPv6 address space for the content publisher is done, and that is used as a prefix for the IPv6 address. For the second 64-bits, an MD5 hash of the content name is done, and the first 64-bits of the hash are converted into the last "quads" of the IPv6 address. Once a content name request is made and published, it is registered with a central registry so that a reverse lookup of the IPv6 address can be easily performed. More details about the architecture are in the poster slide.



"I want http://provider.com/index.html ... which references image.jpg"

1. **Determine IPv6 address space of Provider.com**
8079:ef09:ffbc:d6ce:xxxx:xxxx:xxxx:xxxx
2. **Determine IPv6 addresses of content**
/index.html = 8079:ef09:ffbc:d6ce:eacf:331f:ffc:35d4
/image.jpg = 8079:ef09:ffbc:d6ce:d5b:1c4c:7f72:f69
3. **Issue HTTP request to IPv6 address**
http://8079:ef09:ffbc:d6ce:eacf:331f:ffc:35d4/
http://8079:ef09:ffbc:d6ce:d5b:1c4c:7f72:f69/

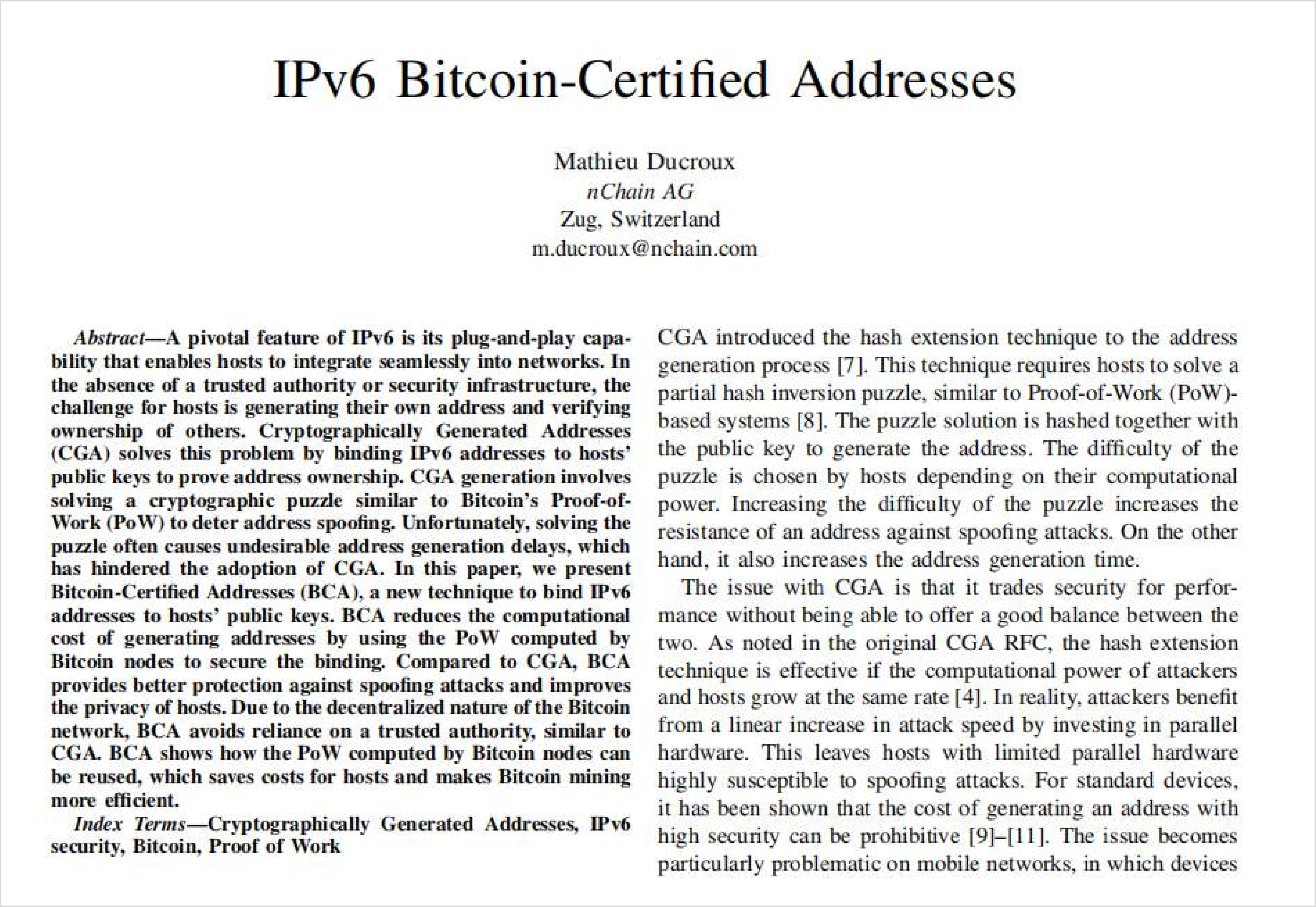
Reverse lookup of IPv6 address. If no content:
(1) Pass request upstream
or
(2) Issue a regular HTTP request, cache it.
This will involve creating a new IPv6 address on a virtual interface.

Handle requests for
http://8079:ef09:ffbc:d6ce:eacf:331f:ffc:35d4/
http://8079:ef09:ffbc:d6ce:d5b:1c4c:7f72:f69/
And return the content initially obtained from provider.com.
(The HTTP server will have to be configured to return content for the IPv6 addresses.)

Returns HTTP content to the browser

The architecture diagram of our IPv6 content addressing system. In our system, the regular browser makes a HTTP request through a proxy, which translates HTTP requests to an IPv6 content addressing system. The request is sent out over the network, until a router on path that has the content responds to the request. The proxy then translates the retrieved content back into a HTTP response to the user's browser.

IPv6 address for everything



IPv6 Bitcoin-Certified Addresses

Mathieu Ducroux
nChain AG
Zug, Switzerland
m.ducroux@nchain.com

***Abstract*—A pivotal feature of IPv6 is its plug-and-play capability that enables hosts to integrate seamlessly into networks. In the absence of a trusted authority or security infrastructure, the challenge for hosts is generating their own address and verifying ownership of others. Cryptographically Generated Addresses (CGA) solves this problem by binding IPv6 addresses to hosts' public keys to prove address ownership. CGA generation involves solving a cryptographic puzzle similar to Bitcoin's Proof-of-Work (PoW) to deter address spoofing. Unfortunately, solving the puzzle often causes undesirable address generation delays, which has hindered the adoption of CGA. In this paper, we present Bitcoin-Certified Addresses (BCA), a new technique to bind IPv6 addresses to hosts' public keys. BCA reduces the computational cost of generating addresses by using the PoW computed by Bitcoin nodes to secure the binding. Compared to CGA, BCA provides better protection against spoofing attacks and improves the privacy of hosts. Due to the decentralized nature of the Bitcoin network, BCA avoids reliance on a trusted authority, similar to CGA. BCA shows how the PoW computed by Bitcoin nodes can be reused, which saves costs for hosts and makes Bitcoin mining more efficient.**

***Index Terms*—Cryptographically Generated Addresses, IPv6 security, Bitcoin, Proof of Work**

CGA introduced the hash extension technique to the address generation process [7]. This technique requires hosts to solve a partial hash inversion puzzle, similar to Proof-of-Work (PoW)-based systems [8]. The puzzle solution is hashed together with the public key to generate the address. The difficulty of the puzzle is chosen by hosts depending on their computational power. Increasing the difficulty of the puzzle increases the resistance of an address against spoofing attacks. On the other hand, it also increases the address generation time.

The issue with CGA is that it trades security for performance without being able to offer a good balance between the two. As noted in the original CGA RFC, the hash extension technique is effective if the computational power of attackers and hosts grow at the same rate [4]. In reality, attackers benefit from a linear increase in attack speed by investing in parallel hardware. This leaves hosts with limited parallel hardware highly susceptible to spoofing attacks. For standard devices, it has been shown that the cost of generating an address with high security can be prohibitive [9]–[11]. The issue becomes particularly problematic on mobile networks, in which devices

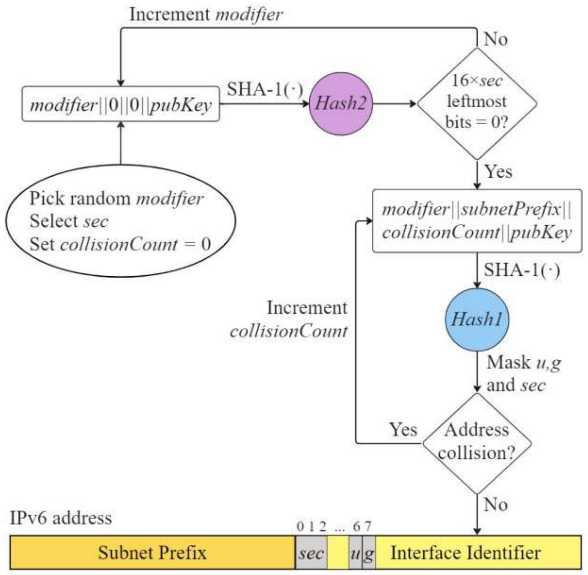


Fig. 1. Detailed data flow of the CGA generation algorithm.

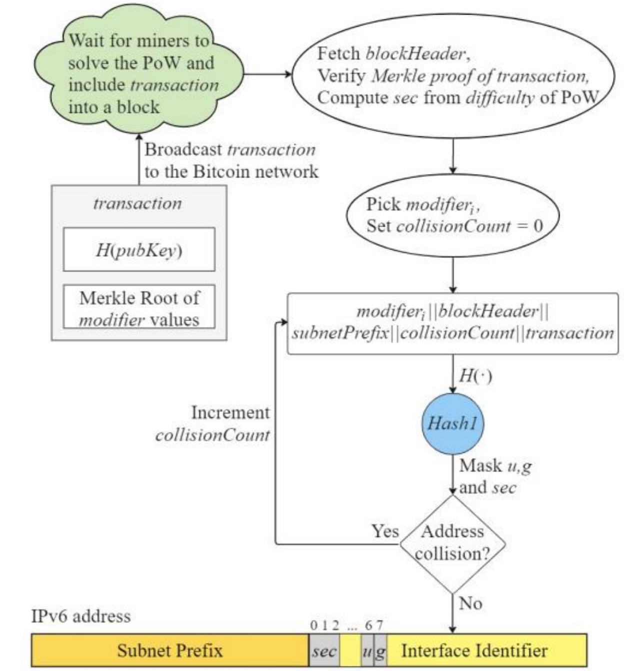
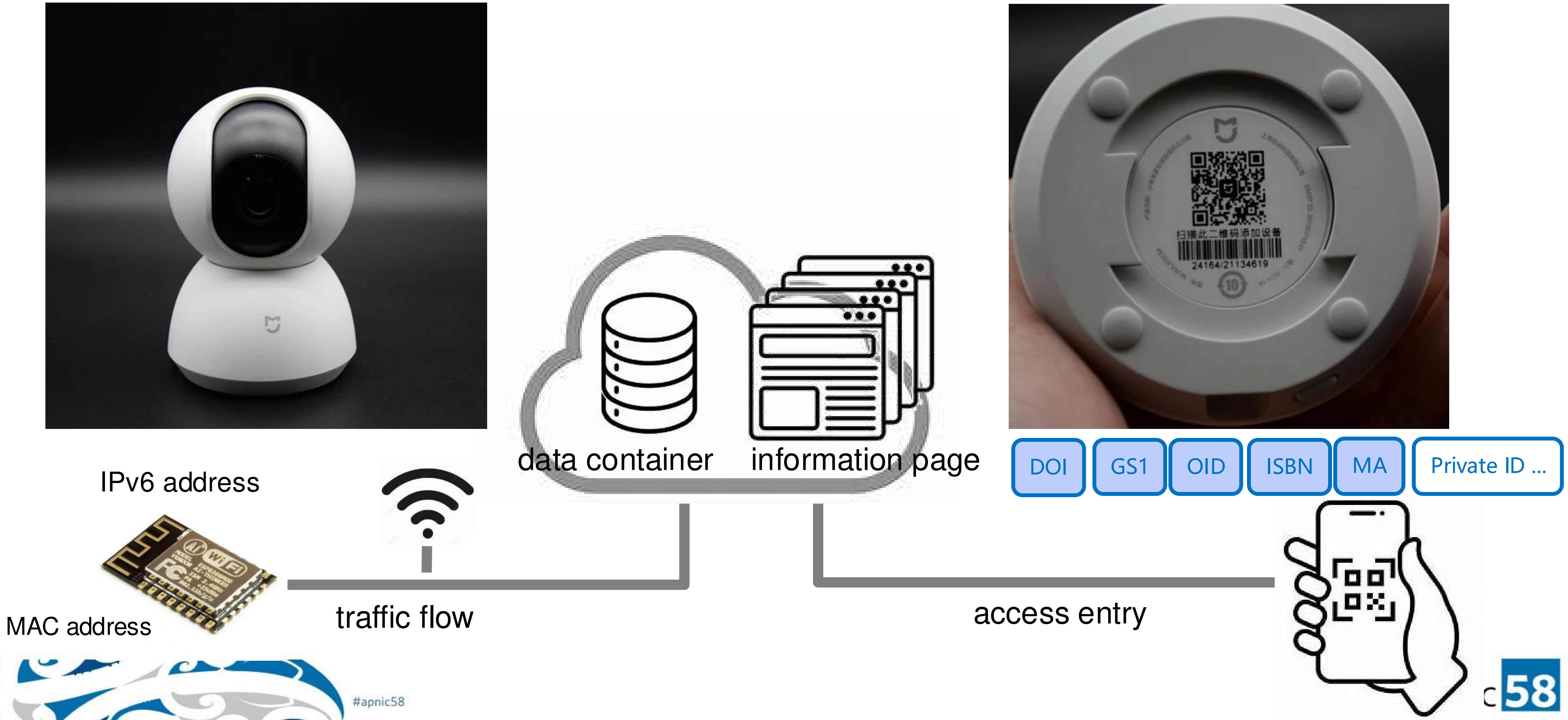

Wait for miners to solve the PoW and include *transaction* into a block
Fetch *blockHeader*,
Verify *Merkle proof of transaction*,
Compute *sec* from *difficulty* of PoW
Broadcast *transaction* to the Bitcoin network
transaction
H(*pubKey*)
Merkle Root of *modifier* values
Pick *modifier*$_i$,
Set *collisionCount* = 0
modifier$_i$||*blockHeader*||*subnetPrefix*||*collisionCount*||*transaction*
H(·)
Increment *collisionCount*
Hash1
Mask *u,g* and *sec*
Yes
Address collision?
No
IPv6 address
0 1 2 ... 6 7
Subnet Prefix
sec
u
g
Interface Identifier

Fig. 2. Detailed data flow of the public key registration process and BCA generation algorithm.

IPv6 address and IoT item Identifiers



IPv6 address

MAC address

traffic flow

data container

information page

access entry

DOI

GS1

OID

ISBN

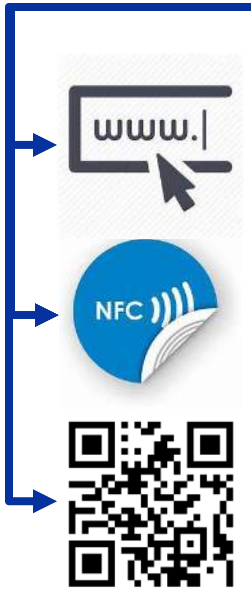
MA

Private ID ...

Challenges to non-electronic item identifiers

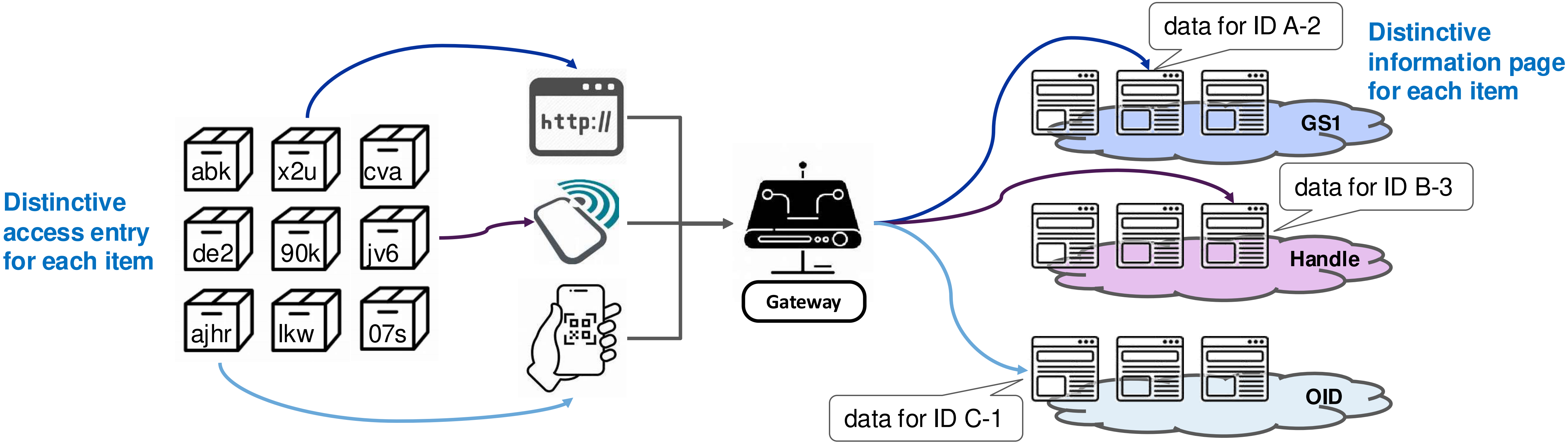
- **Universal Accessibility:**
  - Option1: Relying on dedicated client tools to recognize the semantic and access the info
  - Option2: Assembling the item identifier into a plaintext URL as the access entry, eg:

http://doi.org/10.19363/j.cnki.cn10-1380/tn.2017.10.005



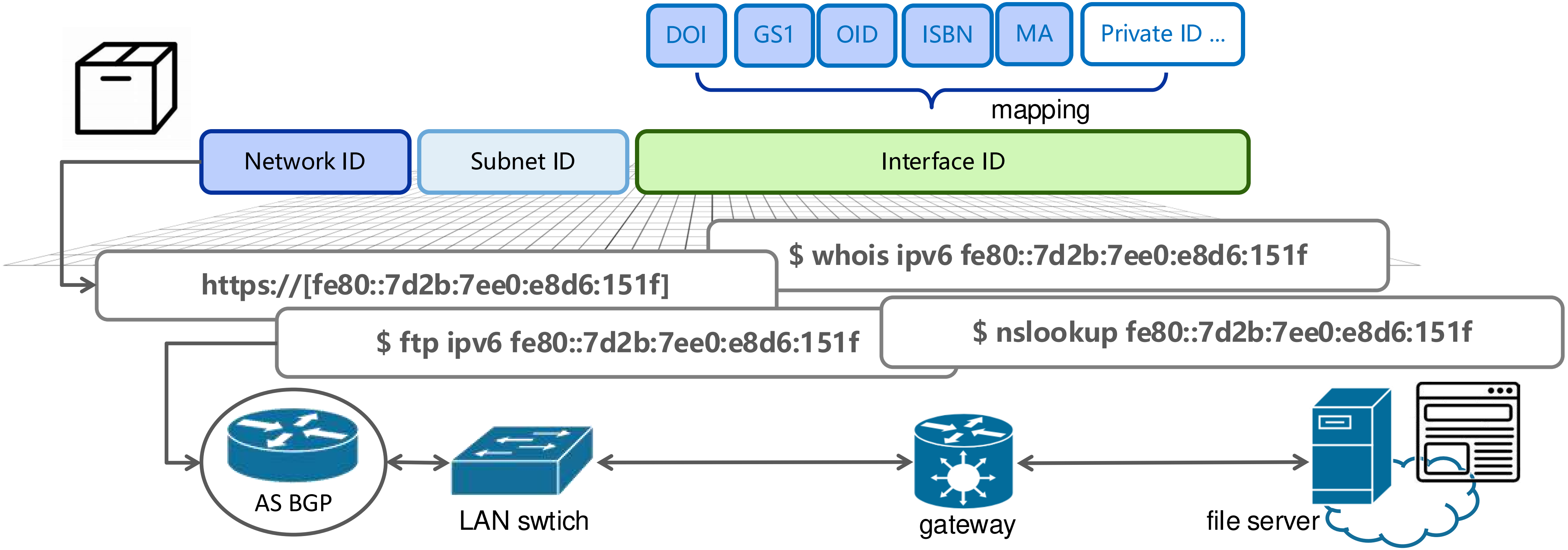
- Identifier counterfeit and web site phishing
- Exposure of the internal semantic encoding rules
- Dependence on external third-party query/whois system
- Compatibility issue among multiple identifiers across diverse systems
- Lack of ability to self-verify authenticity

Local demand for IoT solutions from enterprise customers



- An access query code compatible with all exisitng various types of identifers
- Massive amount of unique random encoding, unique code for each item and information
- An independent query gateway controlled by enterprise and interacted with third-party systems
- Global accessibility regardless the OS and application software
- The ability to self-verify authenticity

IPv6 address to host non-electronic item information



DOI

GS1

OID

ISBN

MA

Private ID …

mapping

Network ID

Subnet ID

Interface ID

https://[fe80::7d2b:7ee0:e8d6:151f]

```
$ whois ipv6 fe80::7d2b:7ee0:e8d6:151f
$ ftp ipv6 fe80::7d2b:7ee0:e8d6:151f
$ nslookup fe80::7d2b:7ee0:e8d6:151f
```

AS BGP

LAN swtich

gateway

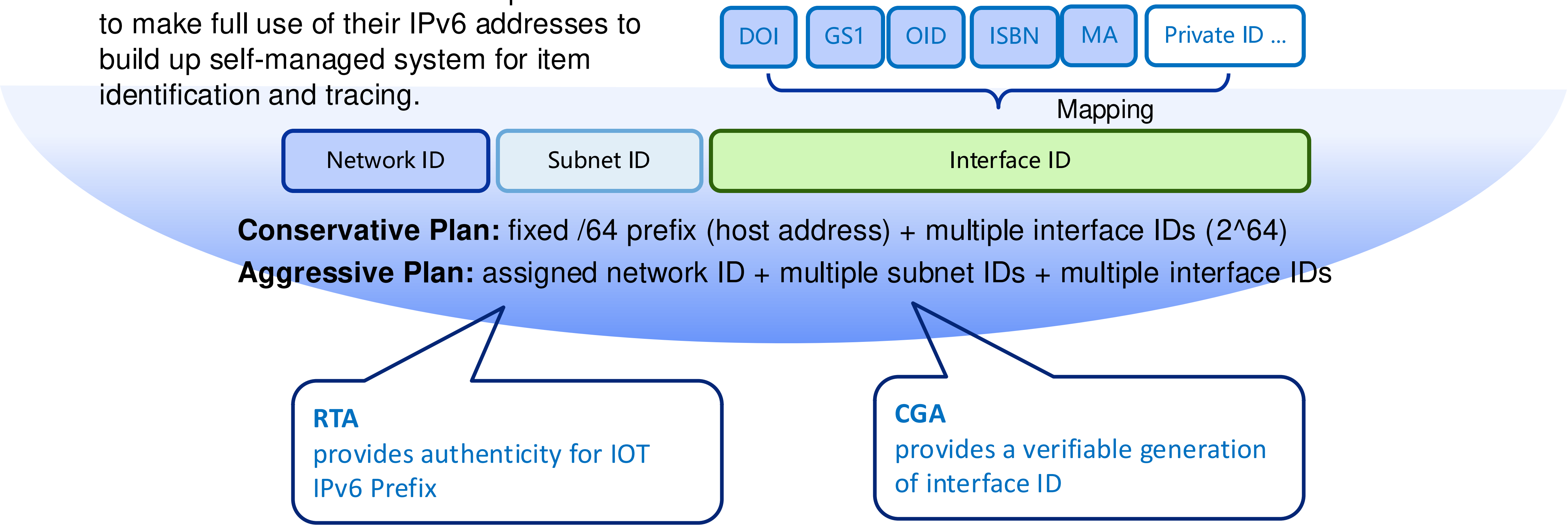
file server

The solution is compliant with the IETF standards framework, and has been deployed by local customers on their IPv6 addresses in both experimental and live network environments.

- Massive non-sequential addresses
- Address == Acdess Identification
- Routing == Verification
- Query flow == traffic flow
- Packet redirected to arbitrary upper layer ID system or app system
- Global acceptance and reachability

Security-enhanced Tech for IPv6 address

The solution enables IPv6 enterprise users to make full use of their IPv6 addresses to build up self-managed system for item identification and tracing.



Conservative Plan: fixed /64 prefix (host address) + multiple interface IDs (2^64)

Aggressive Plan: assigned network ID + multiple subnet IDs + multiple interface IDs

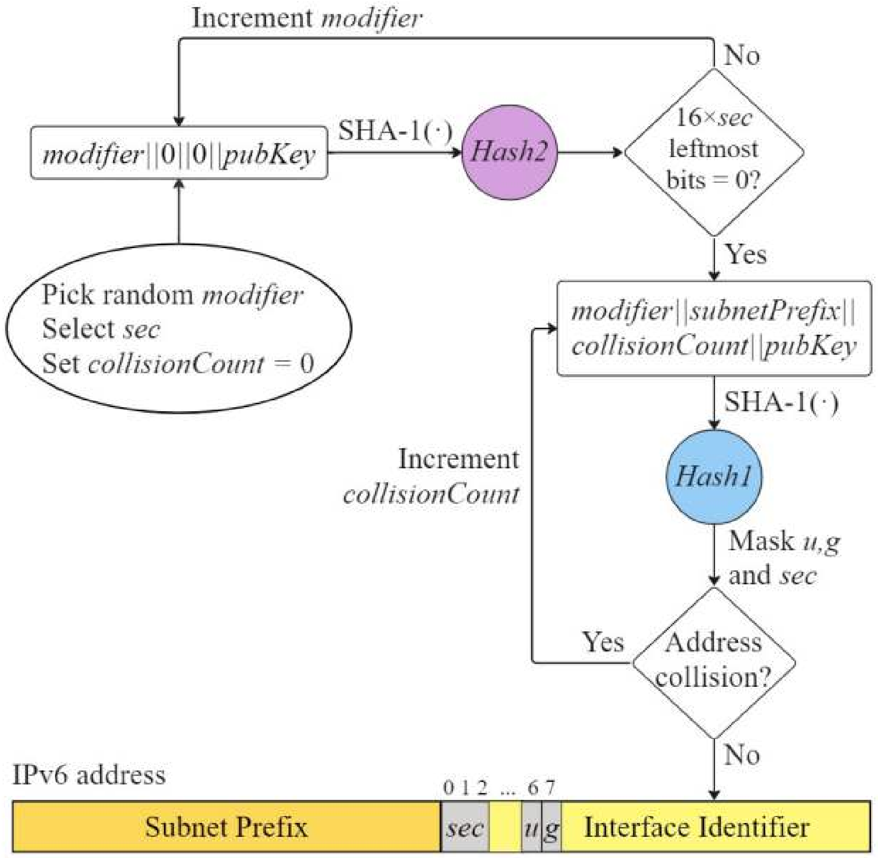
RTA
provides authenticity for IOT IPv6 Prefix

CGA
provides a verifiable generation of interface ID

The technolgoies enhance the security and trustworthiness for IPv6 addresses generated by local enterprise users.

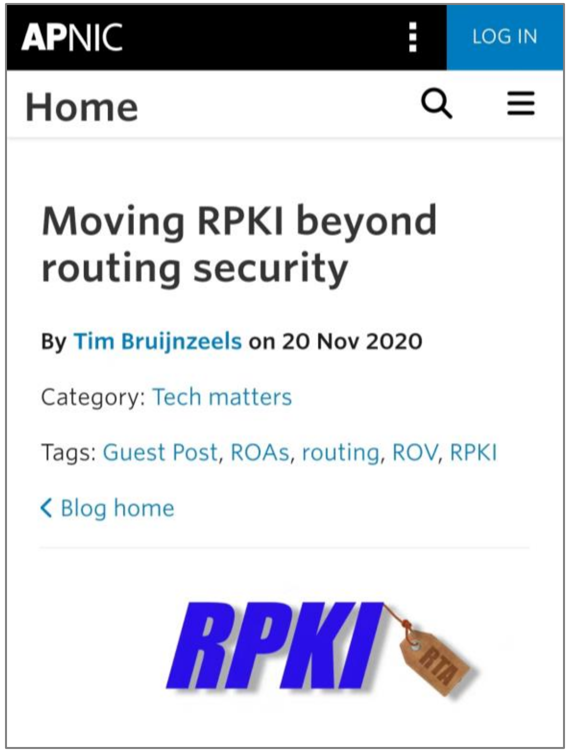
CGA provides a trustful generation of interface ID

RFC 3972 - Cryptographically Generated Addresses (CGA)



- CGA was created to bind a public signature key to an IPv6 address in the Secure Neighbor Discovery Protocol.
- We tentatively employ the original IoT item ID as the modifier parameter, along with other parameters, to generate IoT IPv6 address by CGA generation method.
- The user’s query for the CGA address is supposed to include an encrypted token derived from the IoT item ID, secured using a private key.

Whois, RPKI and RTA provide authenticity for IPv6



APNIC ⋮ LOG IN

Home

Moving RPKI beyond routing security

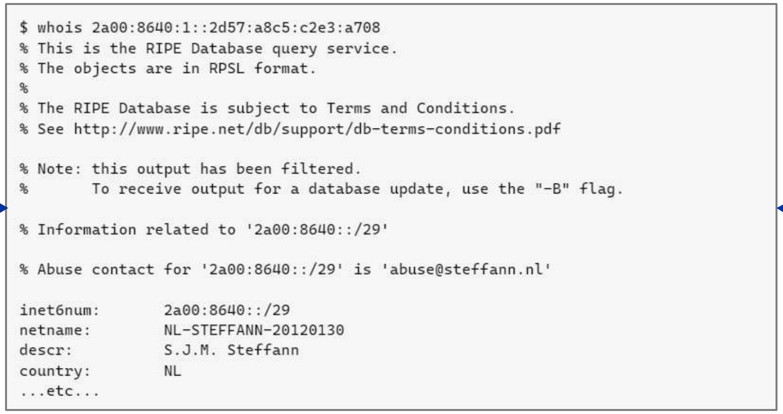
By Tim Bruijnzeels on 20 Nov 2020

Category: Tech matters

Tags: Guest Post, ROAs, routing, ROV, RPKI

‹ Blog home

Resource Tagged Attestations, or RTAs, are a new type of **RPKI** object, allowing any arbitrary file to be signed 'with resources' by one or more parties.
The RTA object is a separate file that cryptographically connects the document with a set of resources. The receiver of the object can use an RPKI validator to show these resources, and verify that it was created by the rightful holder(s) of those resources.



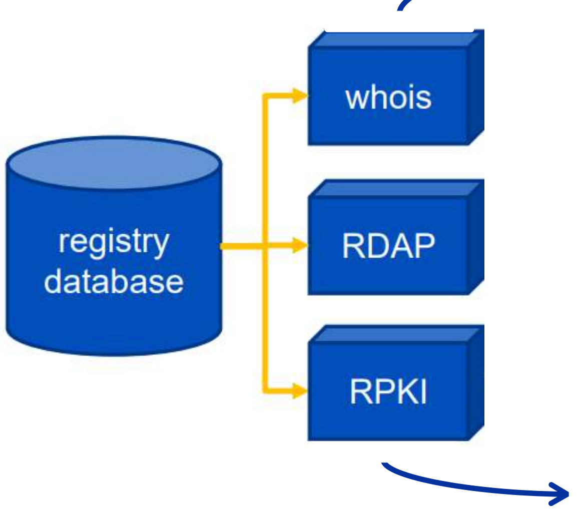
```
$ whois 2a00:8640:1::2d57:a8c5:c2e3:a708
% This is the RIPE Database query service.
% The objects are in RPSL format.
%
% The RIPE Database is subject to Terms and Conditions.
% See http://www.ripe.net/db/support/db-terms-conditions.pdf

% Note: this output has been filtered.
%       To receive output for a database update, use the "-B" flag.

% Information related to '2a00:8640::/29'

% Abuse contact for '2a00:8640::/29' is 'abuse@steffann.nl'

inet6num:       2a00:8640::/29
netname:        NL-STEFFANN-20120130
descr:          S.J.M. Steffann
country:        NL
...etc...
```



IOT IPv6 prefix Meta Data

- IOT usage description
- IOT item lifecycle
- CGA public key
- Necessary graph
- …

APNIC 58

Questions?

#apnic58

TE WHANGANUI A TARA
WELLINGTON,
AOTEAROA NEW ZEALAND
30 August – 6 September 2024