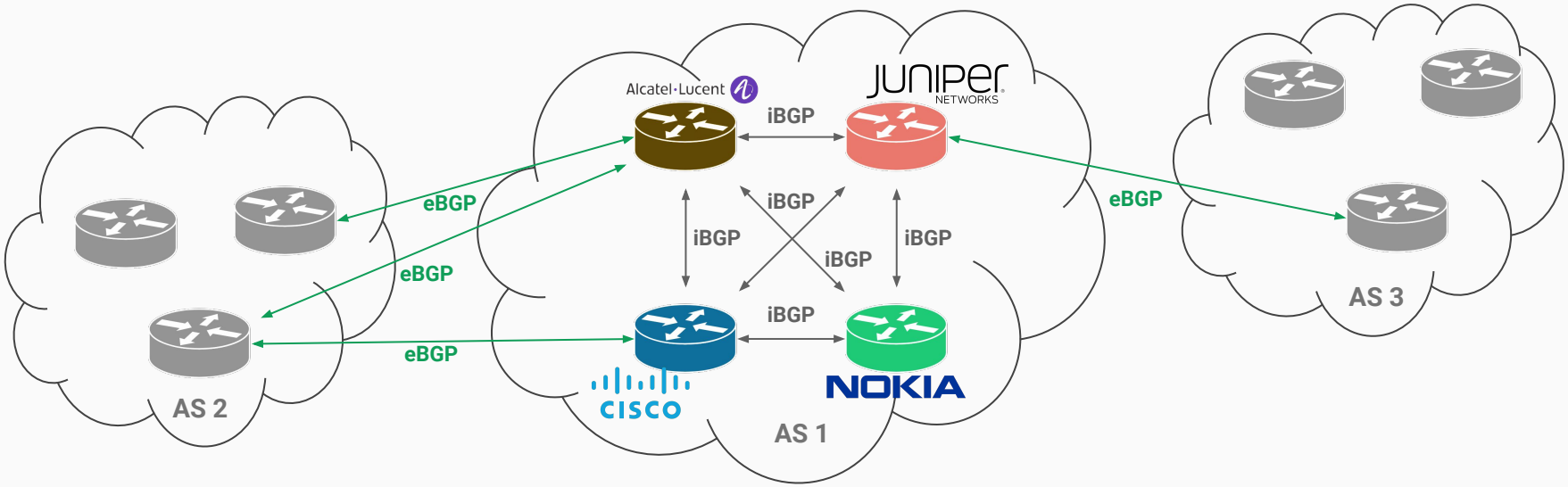
xBGP: Faster Innovation in Routing Protocols

Thomas Wirtgen, Tom Rousseaux, Quentin De Coninck, Nicolas Rybowski, Randy Bush, Laurent Vanbever, Axel Legay, Olivier Bonaventure

Agenda

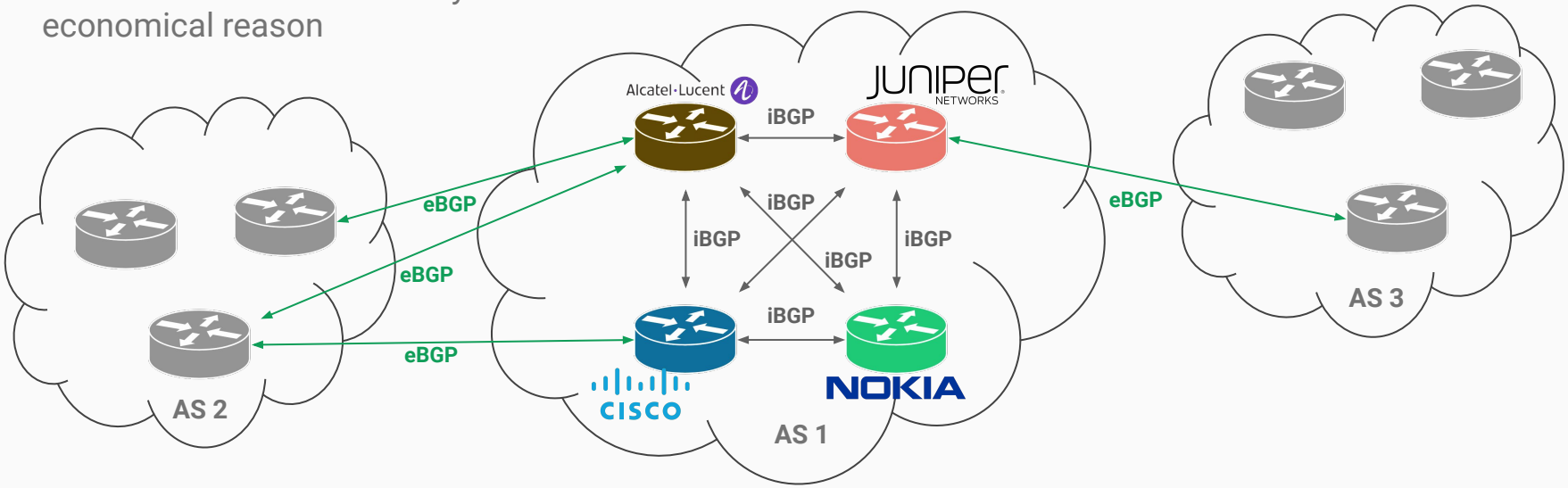
- **Why bringing programmability to BGP?**
- Inside xBGP
- Does using xBGP have an impact on router performances?
- Verifying xBGP extensions
- Conclusion

Routing on the Internet



Routing on the Internet

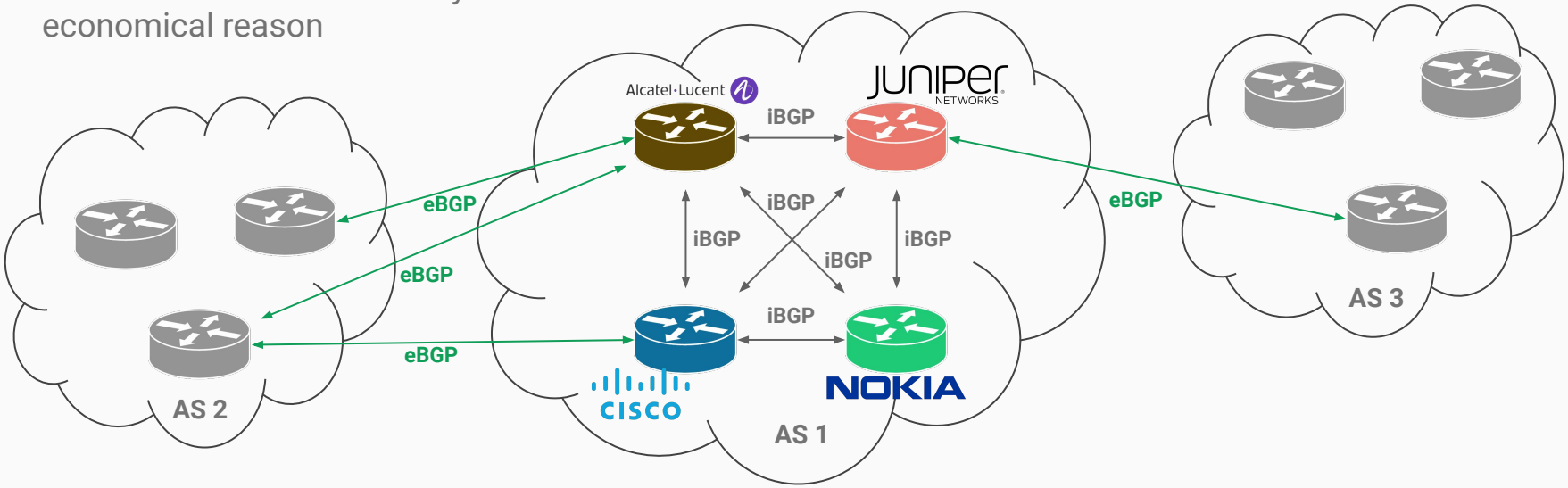
It’s a best practice to have routers from different vendors for stability & economical reason



Routing on the Internet

It’s a best practice to have routers from different vendors for stability & economical reason

All routers do not implement the same set of functionalities



Networks are rapidly evolving

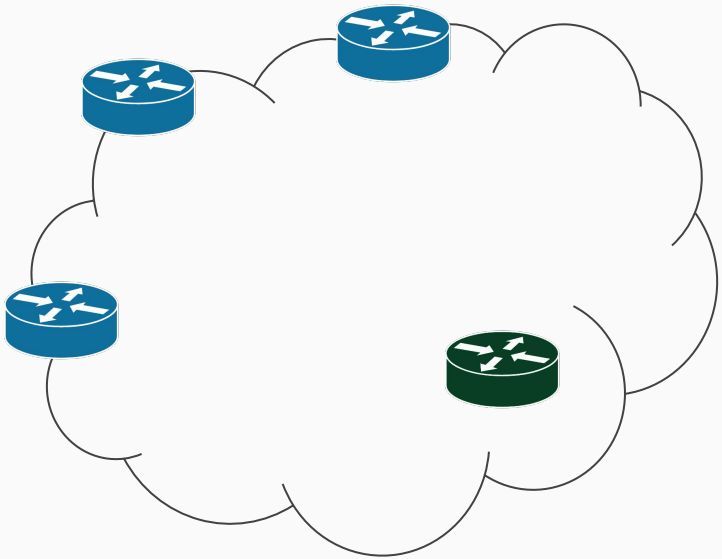
Operators constantly tune their networks

But they are limited:

1. By the Network OS interface (blackbox)
2. By the Standards (BGP + extensions)

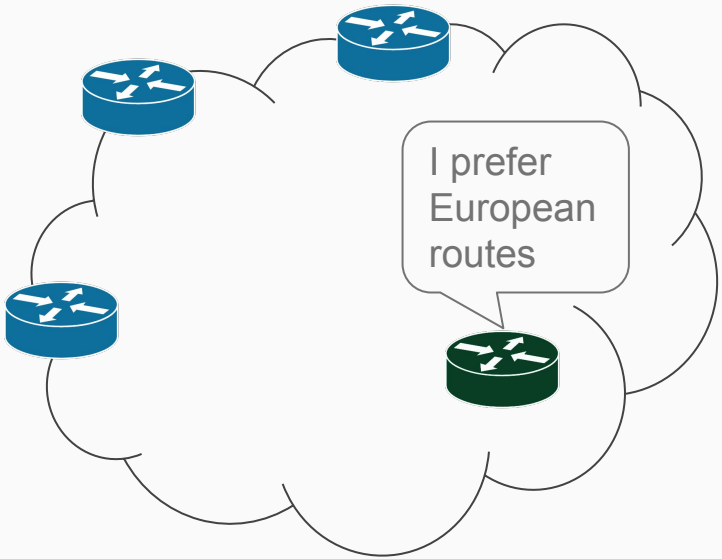
Enhancing the visibility of the BGP control plane

Intra domain routers have no information about the exit router



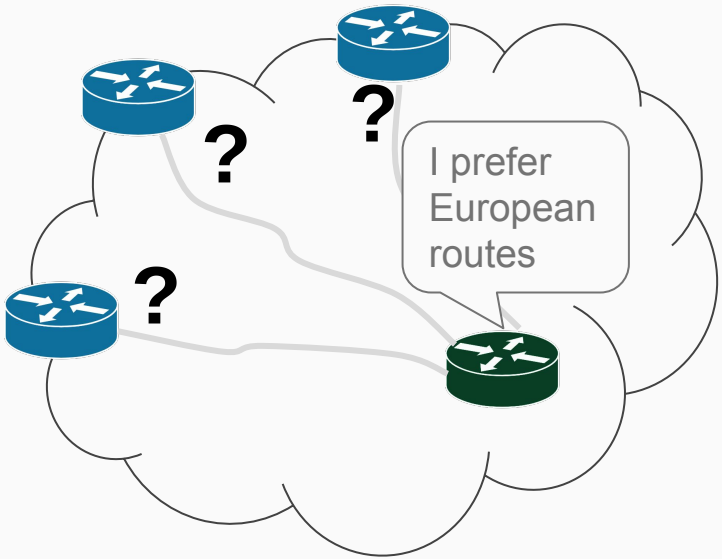
Enhancing the visibility of the BGP control plane

Intra domain routers have no information about the exit router



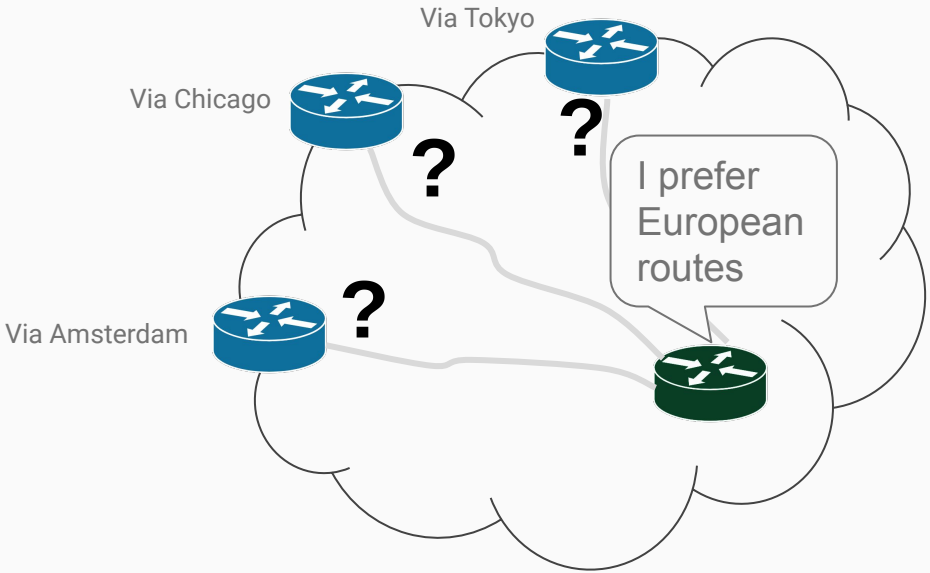
Enhancing the visibility of the BGP control plane

Intra domain routers have no information about the exit router



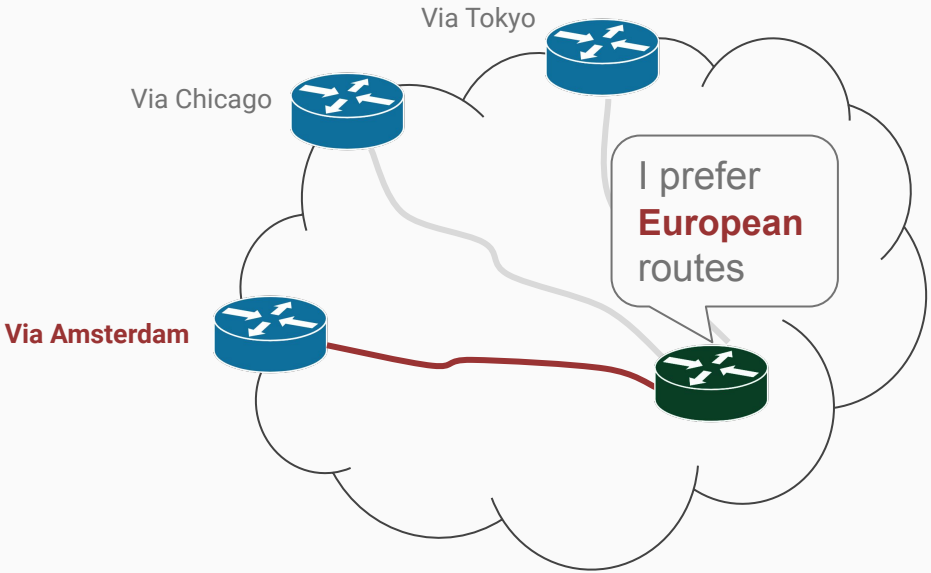
Enhancing the visibility of the BGP control plane

Intra domain routers have no information about the exit router



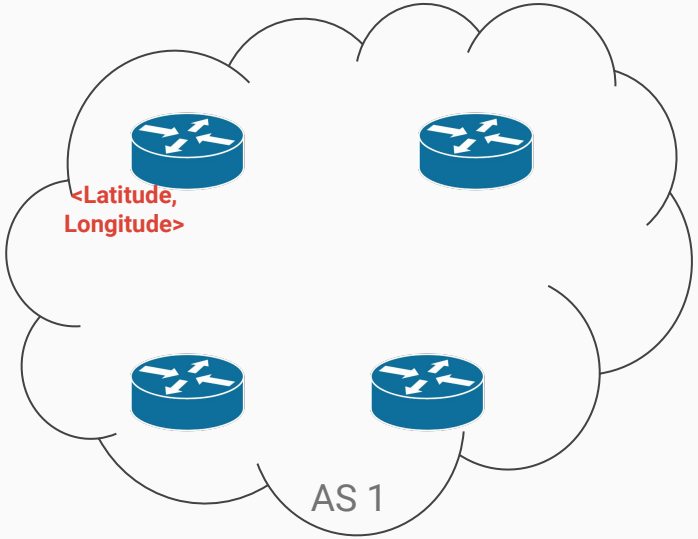
Enhancing the visibility of the BGP control plane

Intra domain routers have no information about the exit router



A complex feature to achieve with classical routers

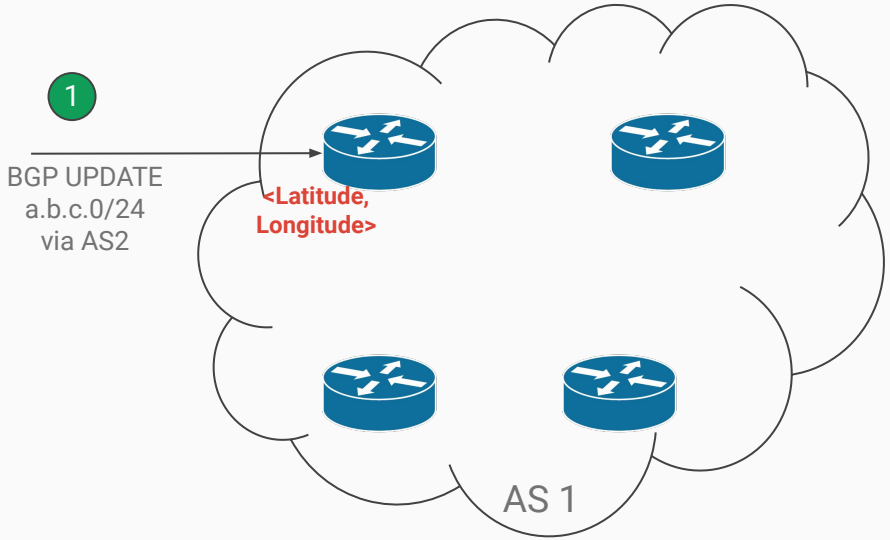
The Geographical Location TLV (GeoLoc TLV)



BGP Path Record Attribute: draft-raszuk-idr-bgp-pr-05

A complex feature to achieve with classical routers

The Geographical Location TLV (GeoLoc TLV)

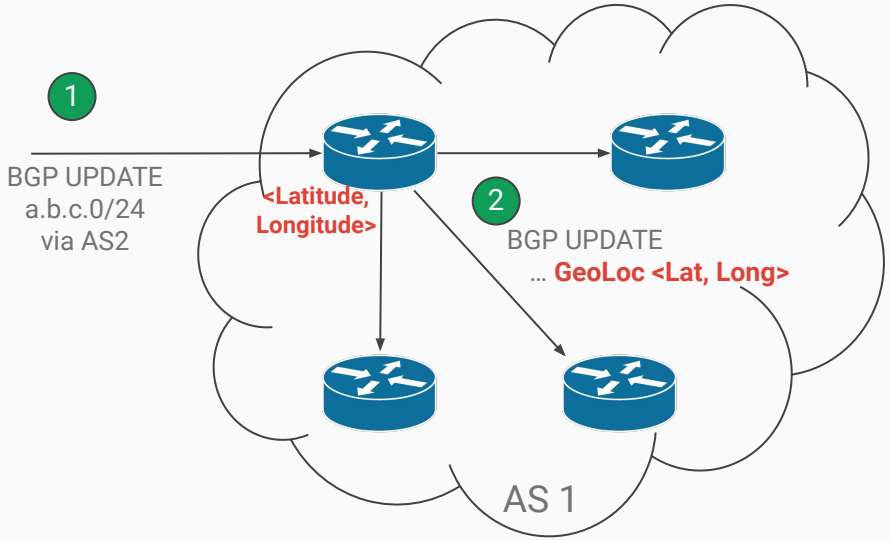


1. Add GeoLoc on the input edge routers

BGP Path Record Attribute: draft-raszuk-idr-bgp-pr-05

A complex feature to achieve with classical routers

The Geographical Location TLV (GeoLoc TLV)

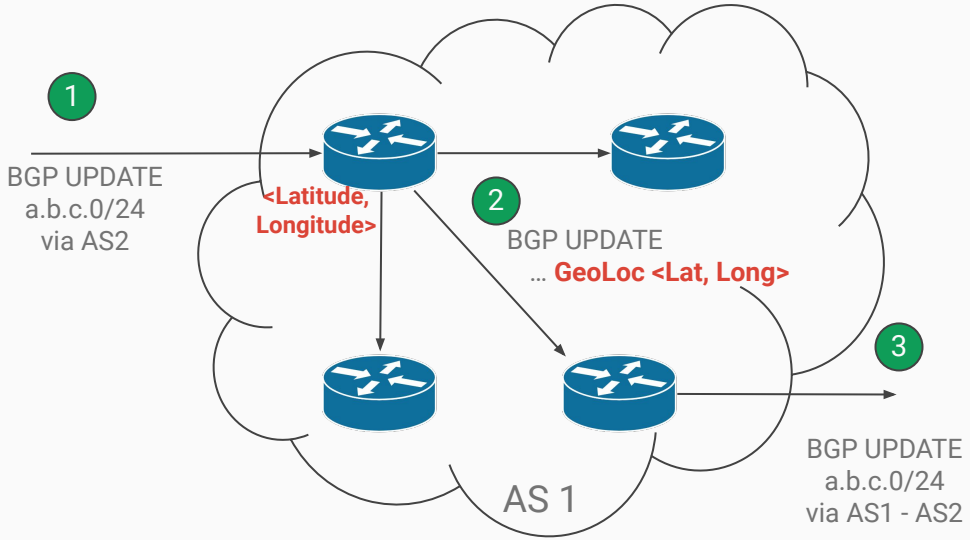


1. Add GeoLoc on the input edge routers
2. Spread the GeoLoc inside the IGP

BGP Path Record Attribute: draft-raszuk-idr-bgp-pr-05

A complex feature to achieve with classical routers

The Geographical Location TLV (GeoLoc TLV)



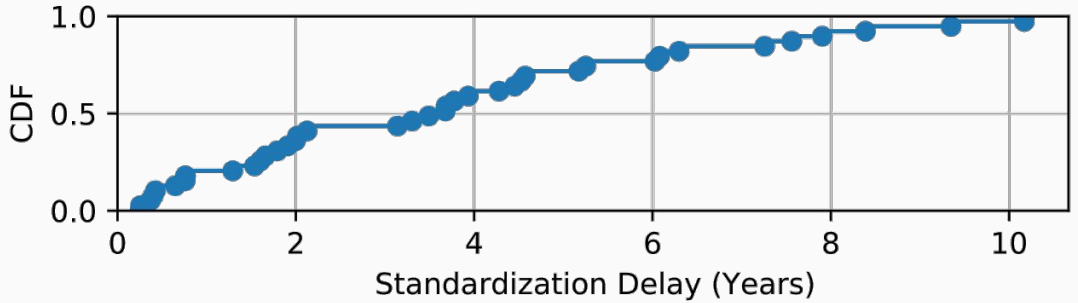
1. Add GeoLoc on the input edge routers
2. Spread the GeoLoc inside the IGP
3. Remove GeoLoc on the output edge routers

BGP Path Record Attribute: draft-raszuk-idr-bgp-pr-05

All that remains is to ship the feature...

One does not simply ask to your routers vendor...

1. Standardisation of the new feature by the IETF
   (3.5 years in average for BGP & confirmed by another study [1])
2. Implementation on the vendor OSes
3. Update your routers



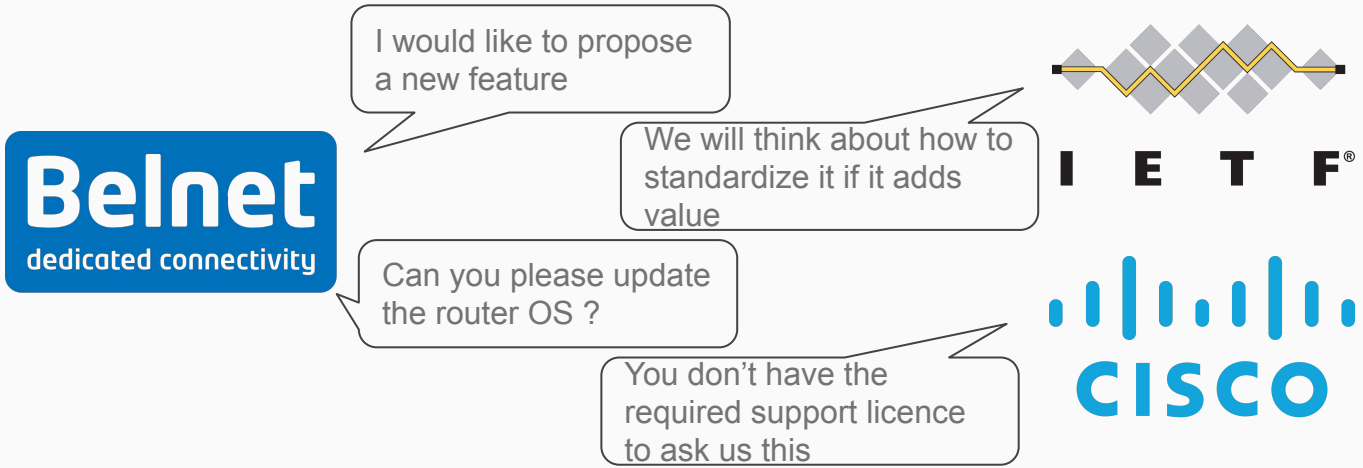
[1] Stephen McQuistin, Mladen Karan, Prashant Khare, Colin Perkins, Gareth Tyson, Matthew Purver, Patrick Healey, Waleed Iqbal, Junaid Qadir, and Ignacio Castro. 2021. Characterising the IETF through the lens of RFC deployment. In Proceedings of the 21st ACM Internet Measurement Conference (IMC '21). Association for Computing Machinery, New York, NY, USA, 137–149. https://doi.org/10.1145/3487552.3487821

All that remains is to ship the feature...

One does not simply ask to your routers vendor...

1. Standardisation of the new feature by the IETF (3.5 years in average for BGP)
2. Implementation on the vendor OSes
3. Update your routers

You can not easily influence steps 1 and 2 !



Current paradigm slows innovation

Problem #1: Routers from different vendors

Problem #2: Protocol extensions not implemented on all routers

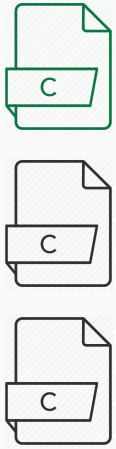
Problem #3: Slow upgrade process

⇒ xBGP is designed to bring innovation & programmability to existing routing protocols

Agenda

- Why bringing programmability to BGP ?
- **Inside xBGP**
- Does using xBGP have an impact on router performances?
- Verifying xBGP extensions
- Conclusion

Classical “update” of routers

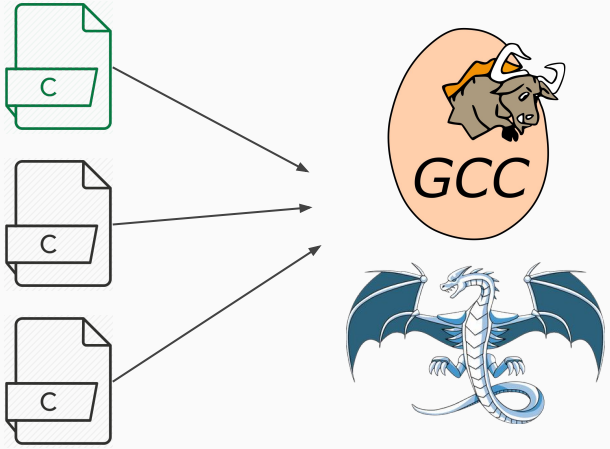


Router operating system source file

Router vendor side

Network operator side

Classical “update” of routers



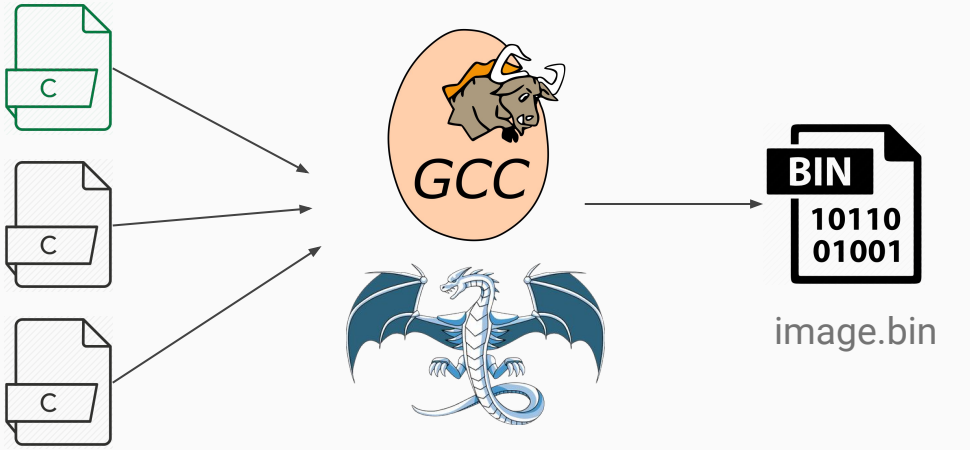
Router operating system source file

Compiler

Router vendor side

Network operator side

Classical “update” of routers



Router operating system source file

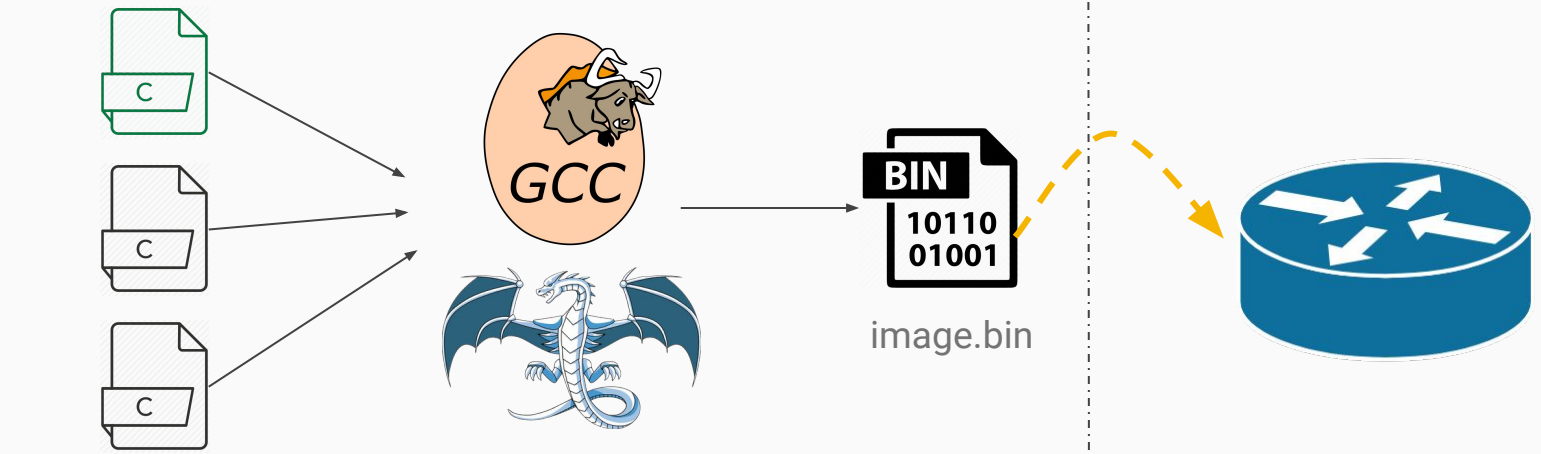
Compiler

Bootable network image

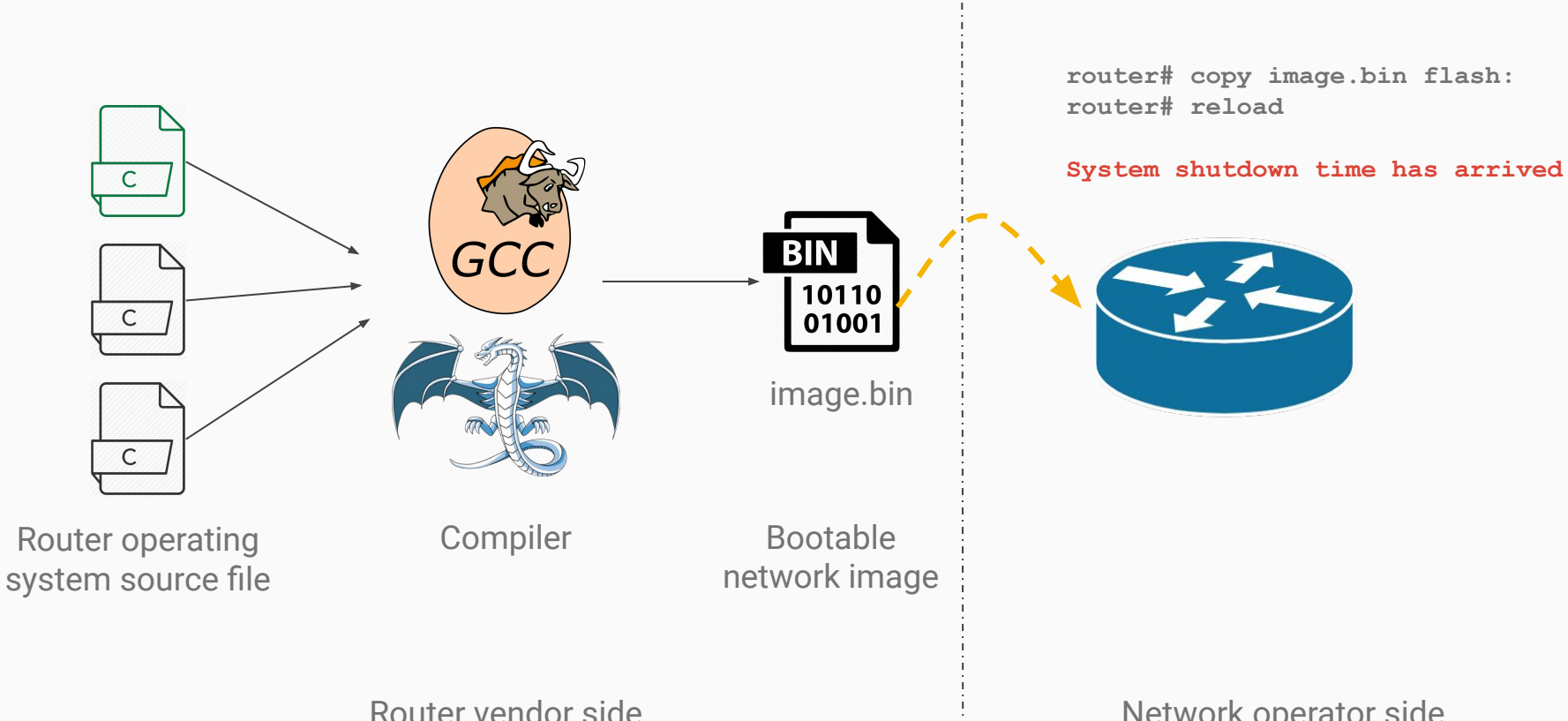
Router vendor side

Network operator side

Classical “update” of routers



Classical “update” of routers



Leveraging eBPF to “bypass” classical updates

Two core components:

Leveraging eBPF to “bypass” classical updates

Two core components:

1. eBPF bytecode
   multi-arch compatible

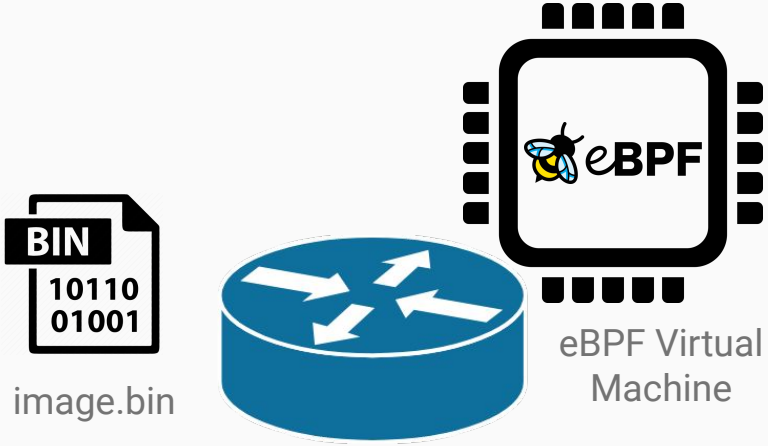
Leveraging eBPF to “bypass” classical updates

Two core components:

1. eBPF bytecode
   multi-arch compatible

2. eBPF runtime environment
   ~ “lightweight JVM” like

Our solution to add new features

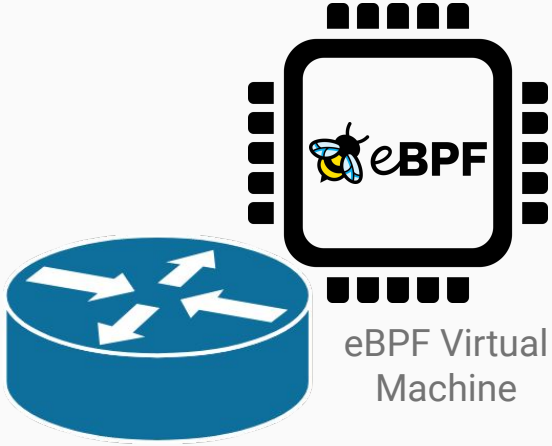


Router vendor side

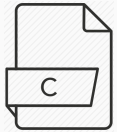
Network operator side

Our solution to add new features

image.bin



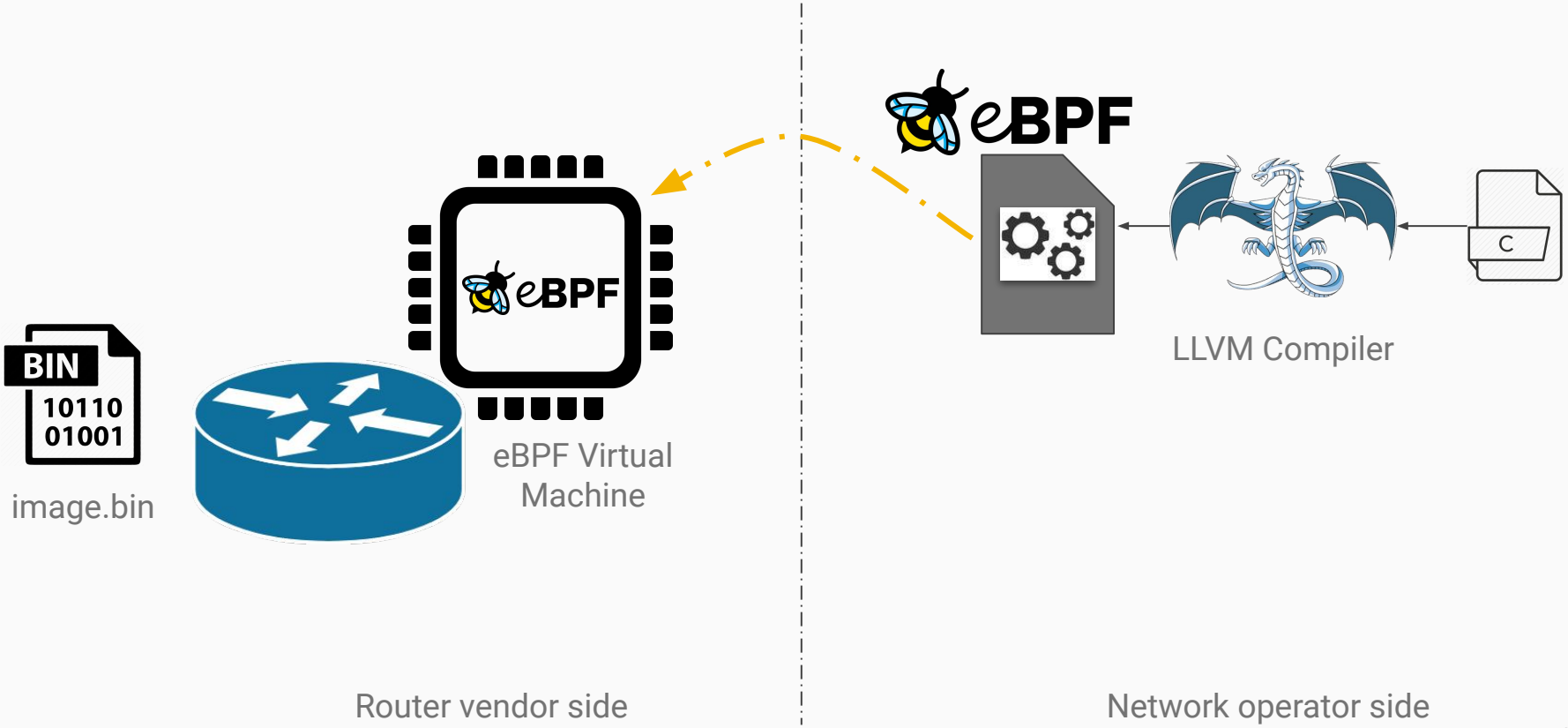
eBPF Virtual Machine



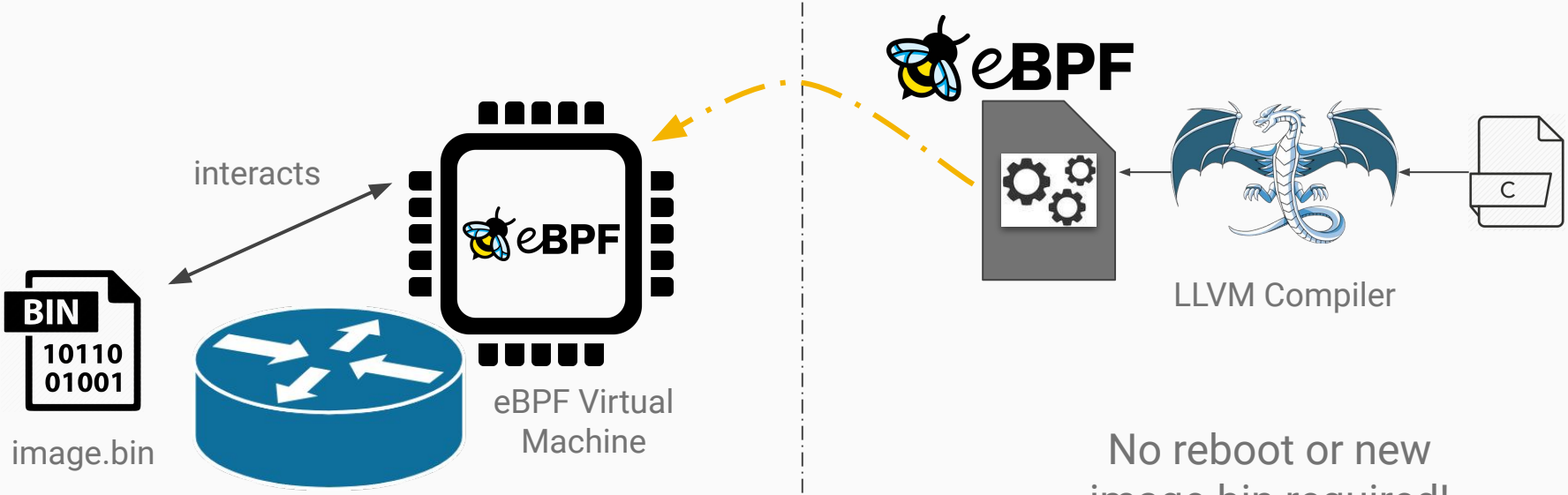
Router vendor side

Network operator side

Our solution to add new features



Our solution to add new features



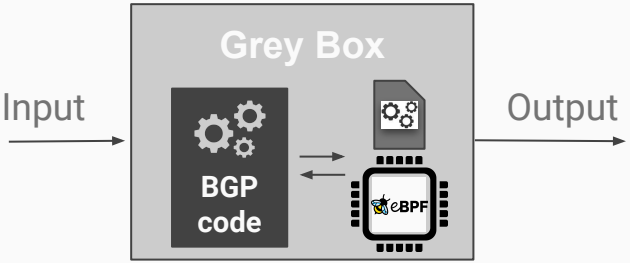
Towards a paradigm shift

From a router vendor **full** control

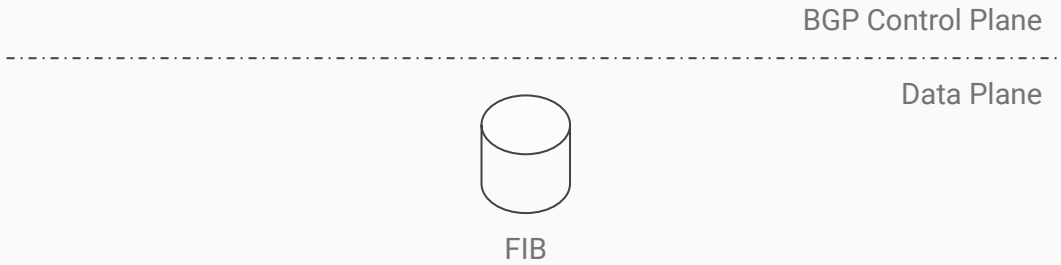
Towards a paradigm shift

From a router vendor **full** control

To the router vendor & **network operator** control

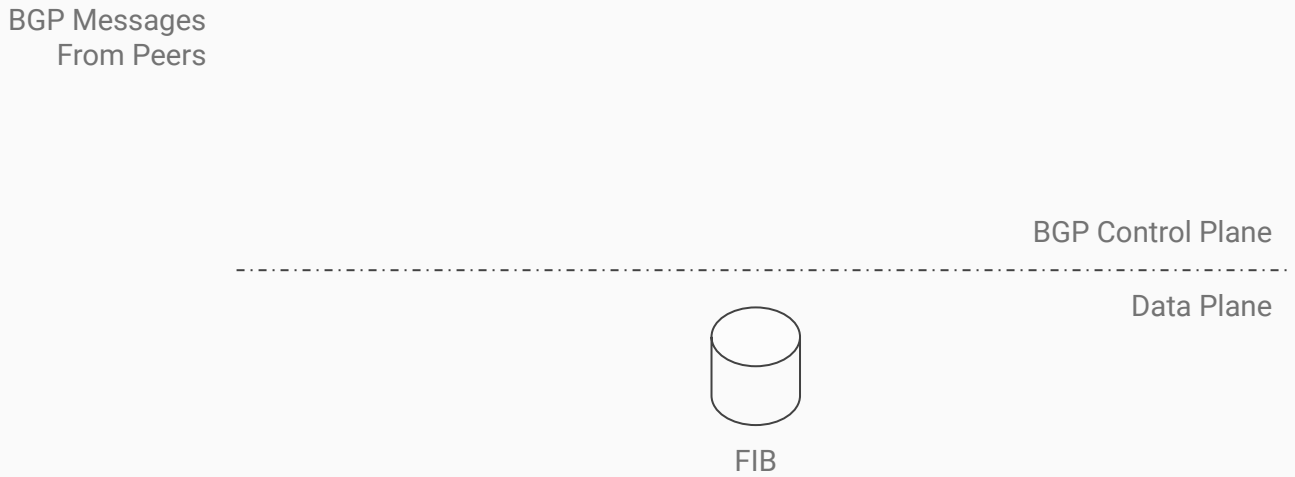


GeoLoc needs to alter the BGP Workflow



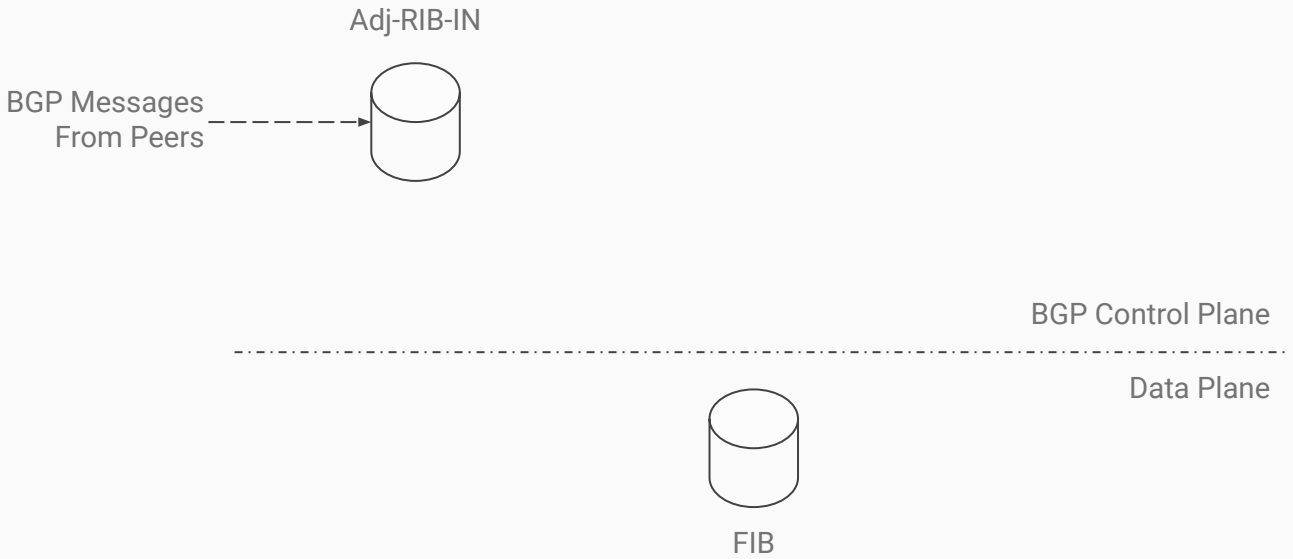
RFC 4271 BGP Workflow

GeoLoc needs to alter the BGP Workflow



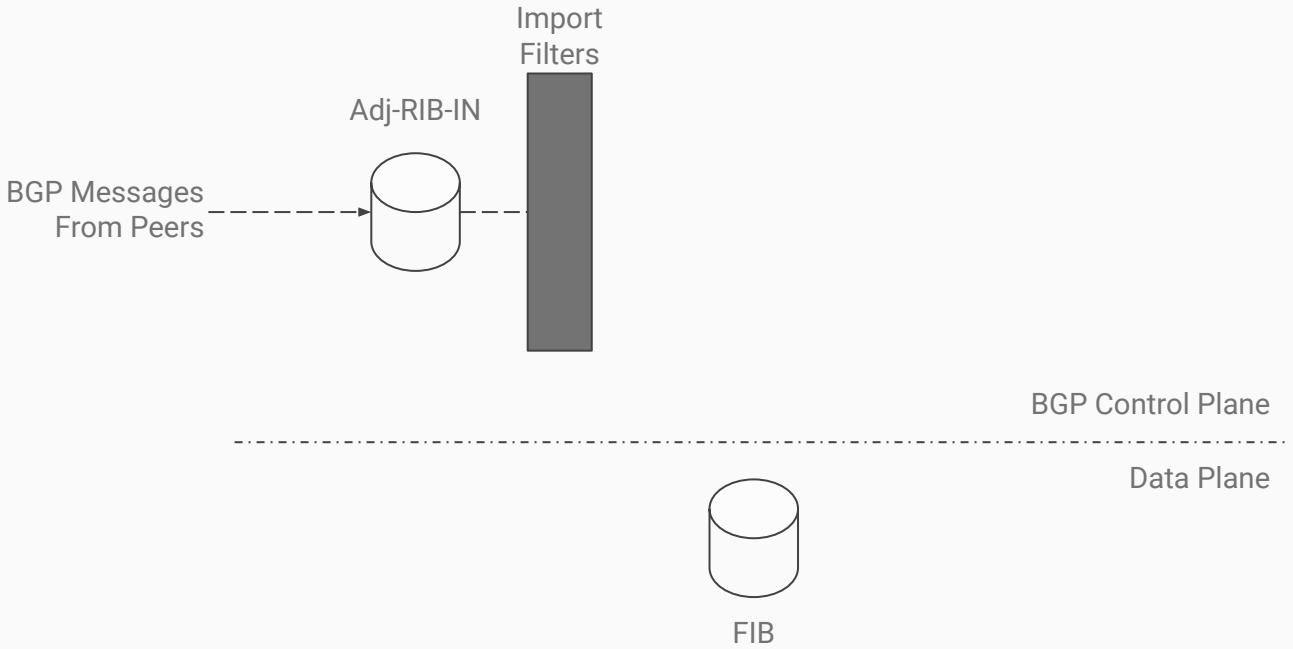
RFC 4271 BGP Workflow

GeoLoc needs to alter the BGP Workflow



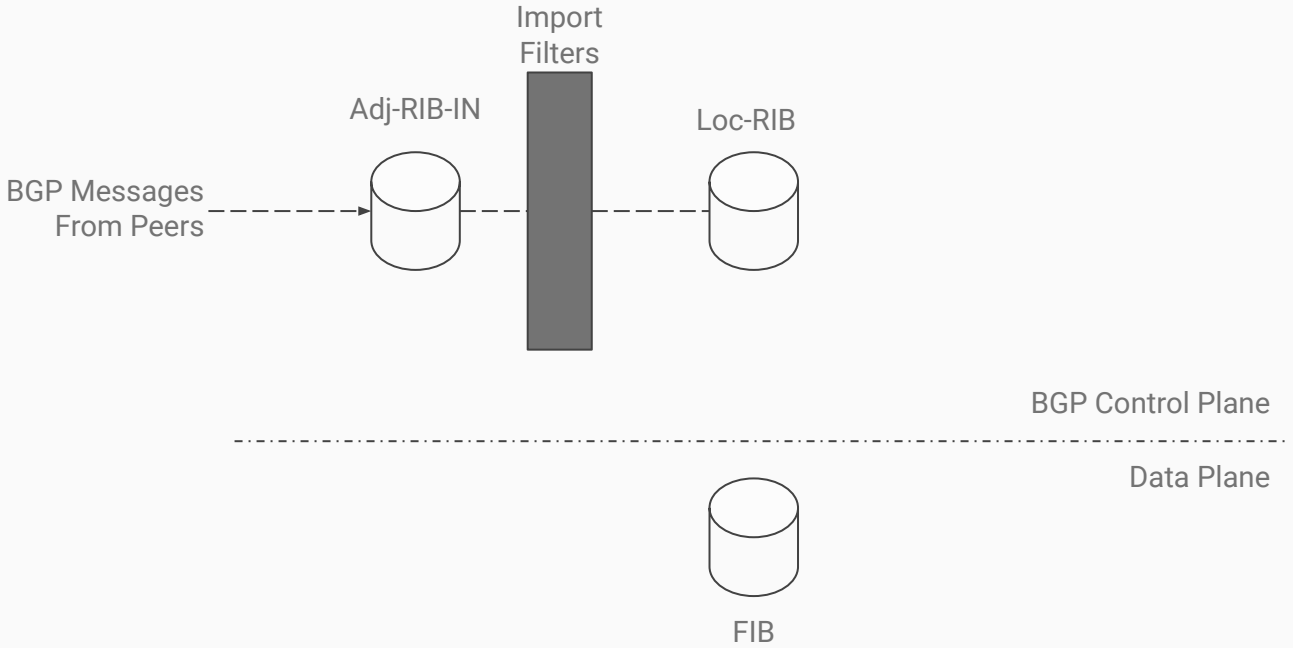
RFC 4271 BGP Workflow

GeoLoc needs to alter the BGP Workflow



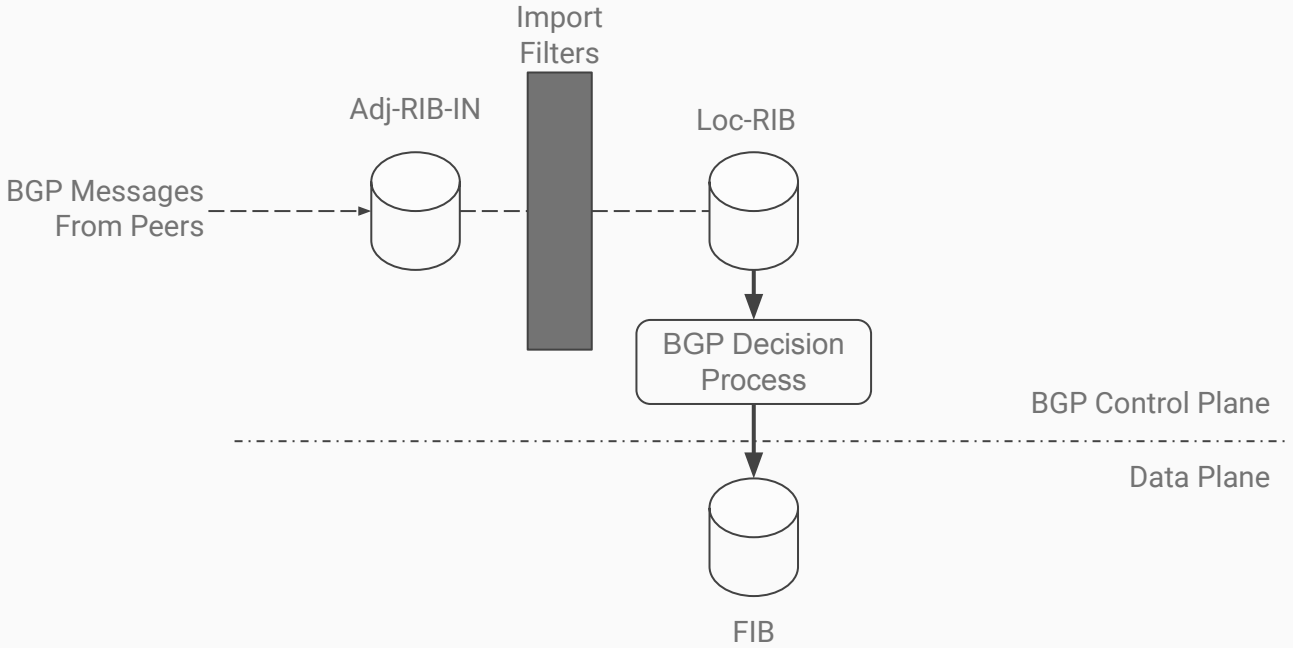
RFC 4271 BGP Workflow

GeoLoc needs to alter the BGP Workflow



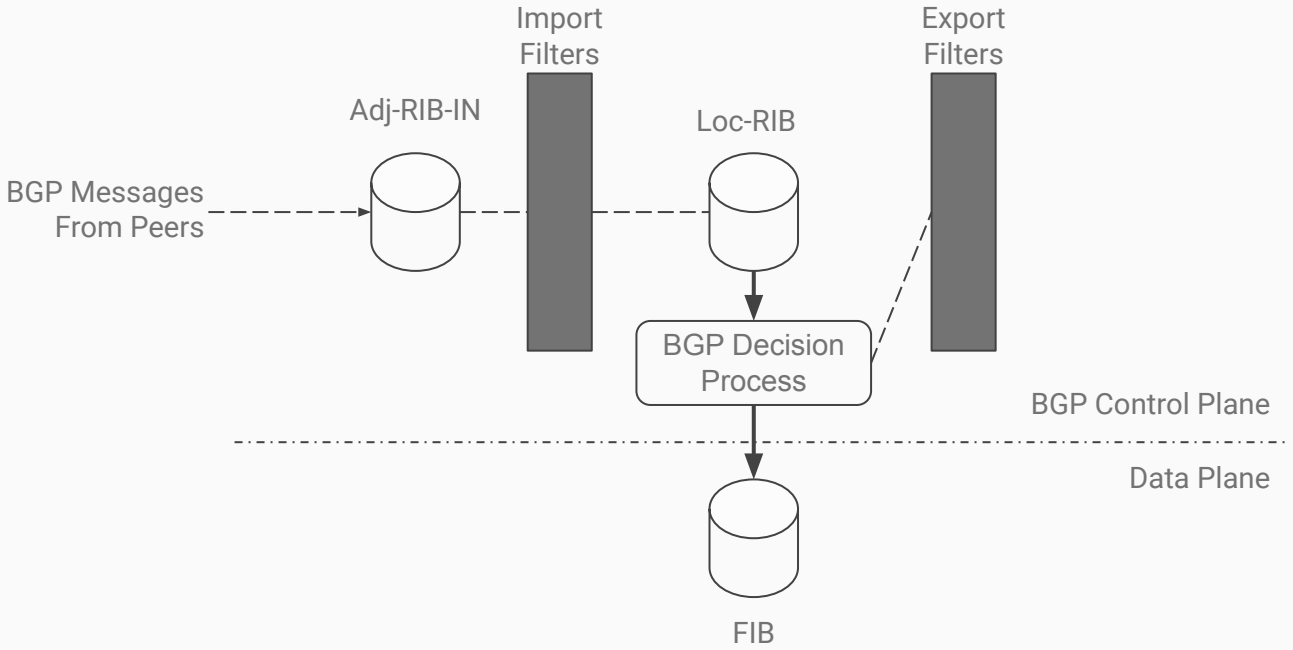
RFC 4271 BGP Workflow

GeoLoc needs to alter the BGP Workflow



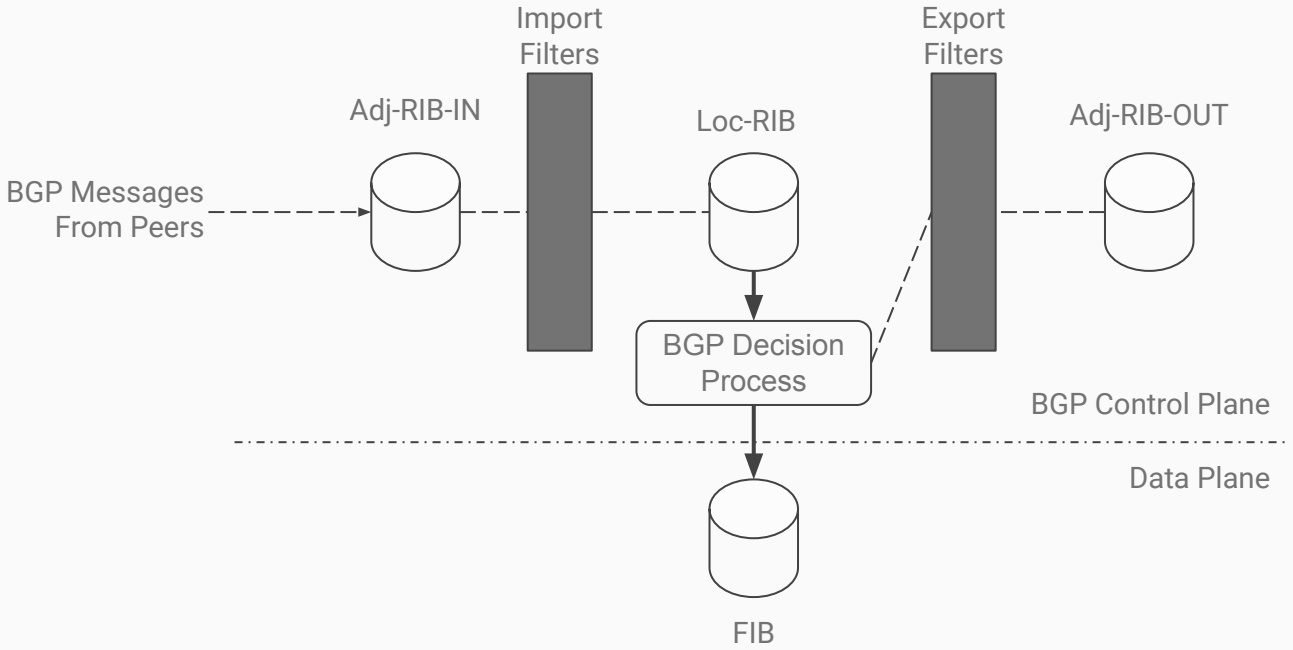
RFC 4271 BGP Workflow

GeoLoc needs to alter the BGP Workflow



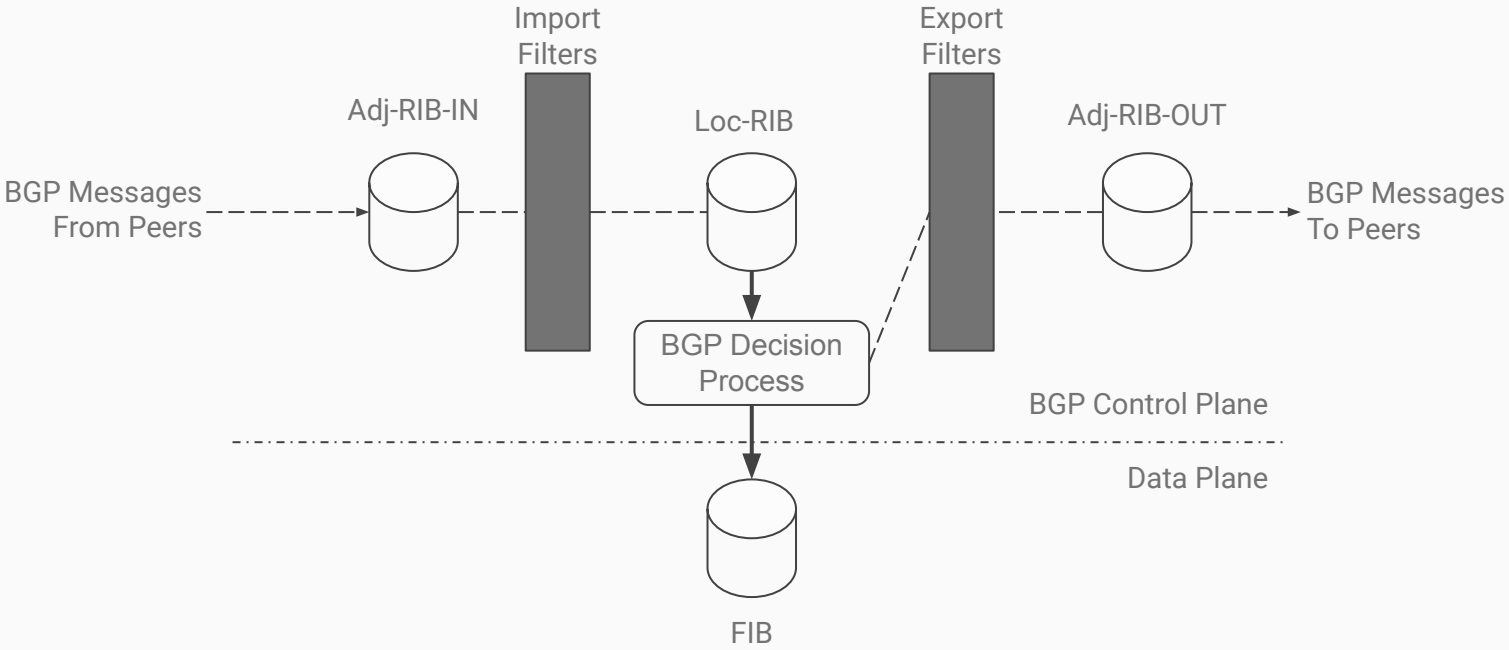
RFC 4271 BGP Workflow

GeoLoc needs to alter the BGP Workflow



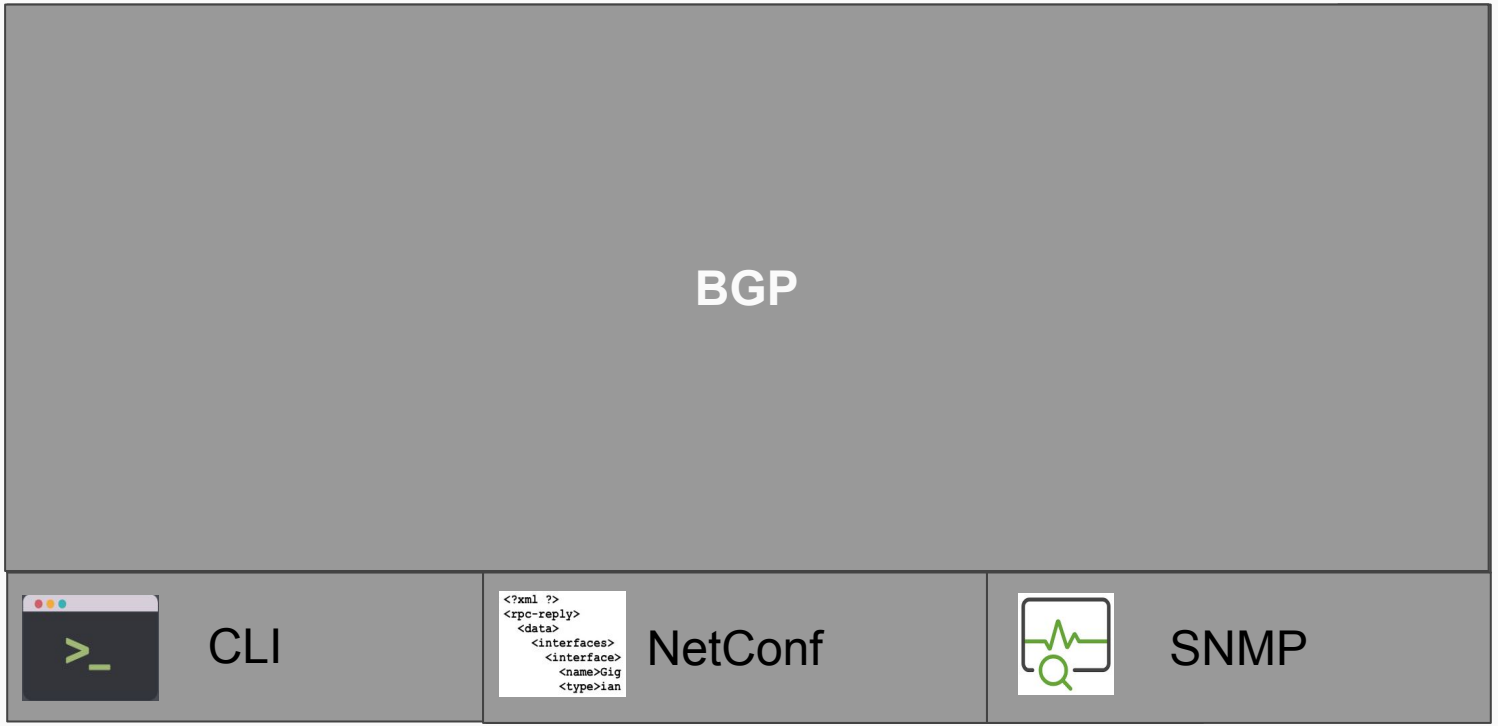
RFC 4271 BGP Workflow

GeoLoc needs to alter the BGP Workflow

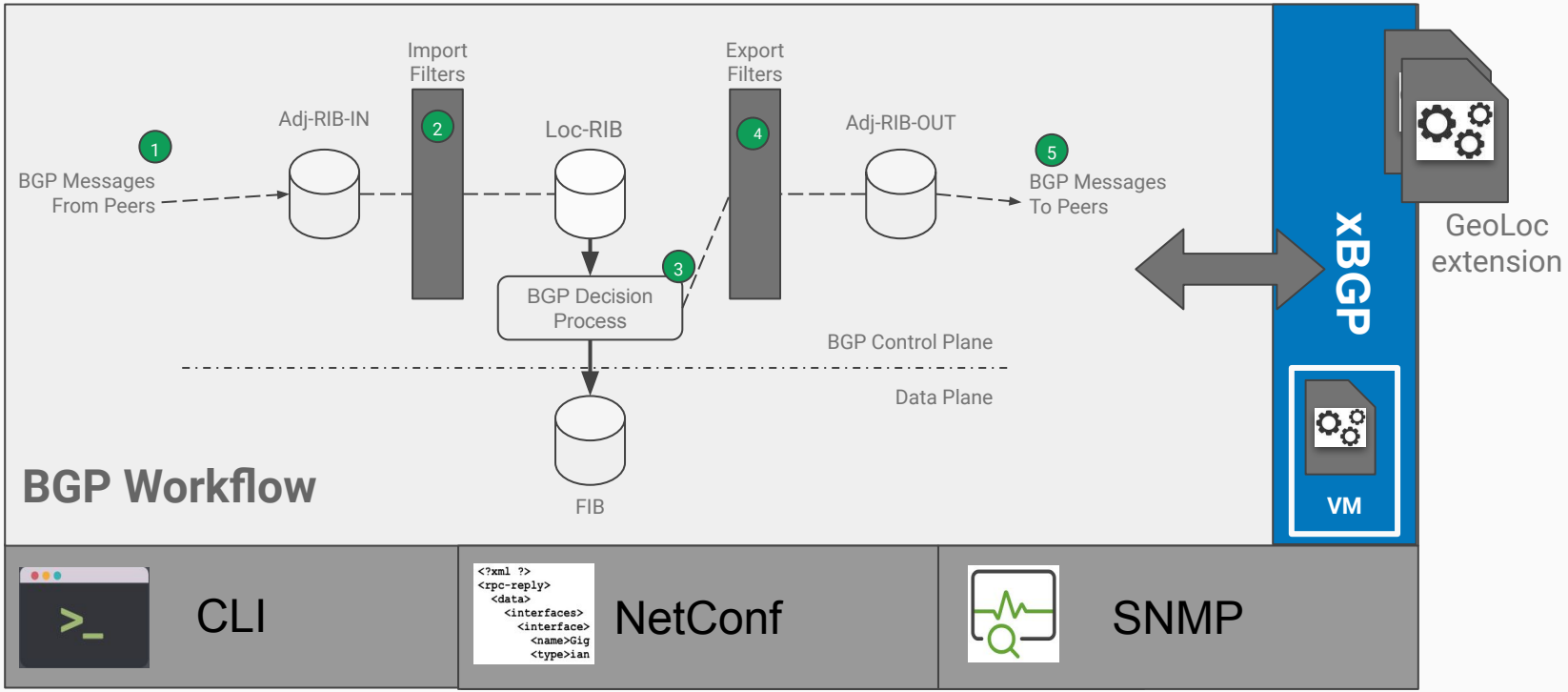


RFC 4271 BGP Workflow

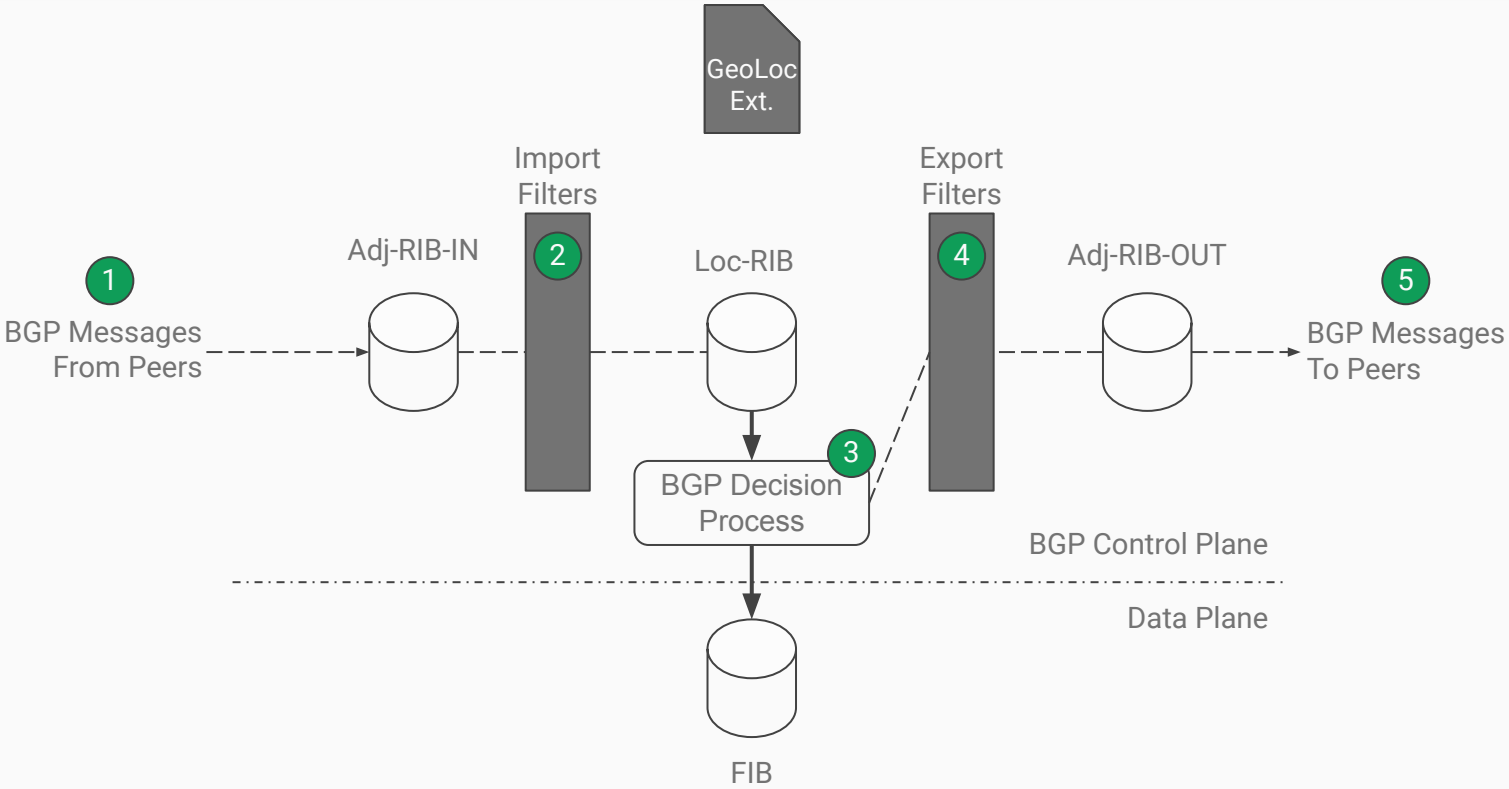
Traditional BGP implementations are opaque



BGP workflow are now **exposed with xBGP**

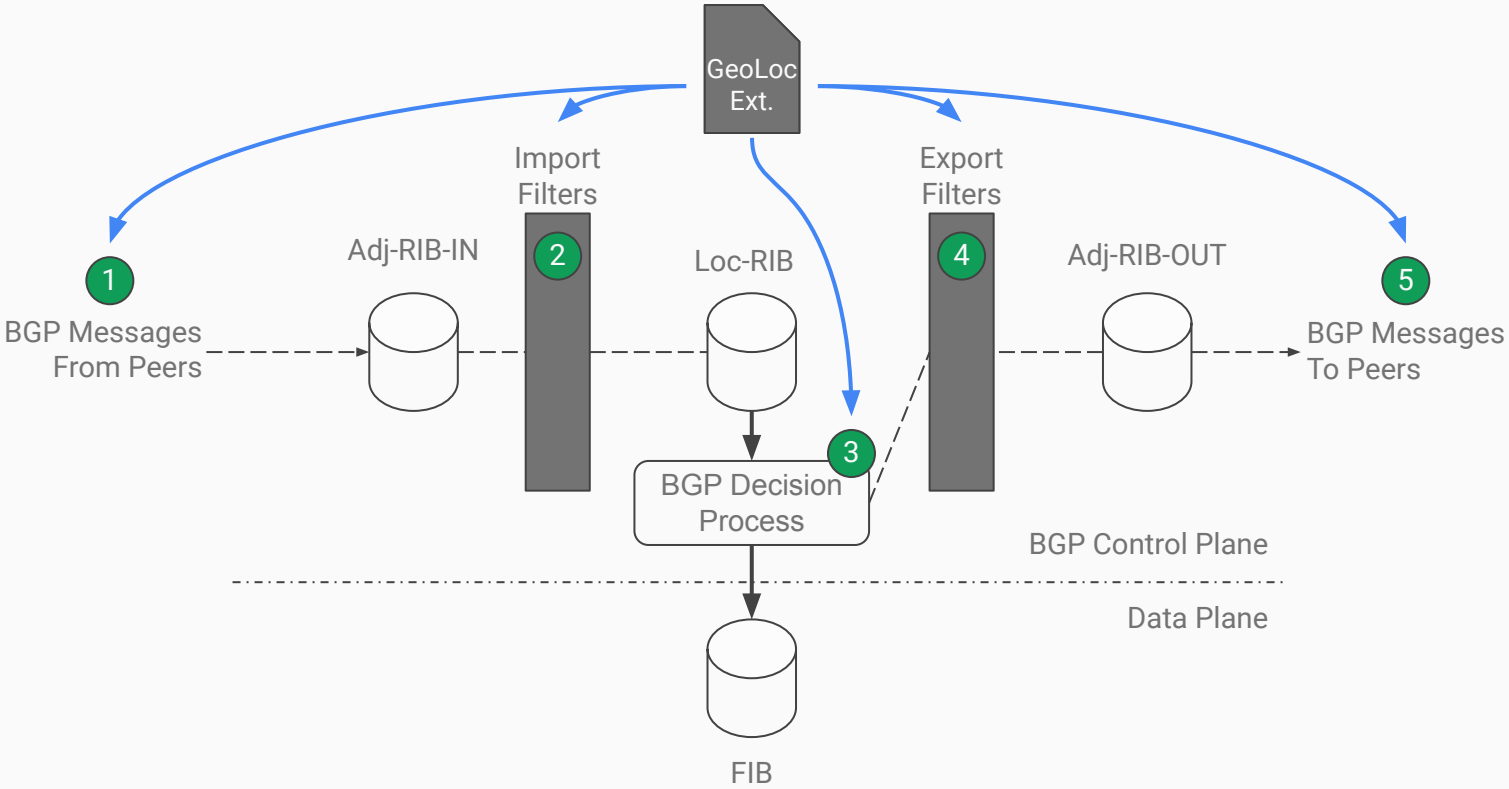


GeoLoc needs to alter the BGP Workflow



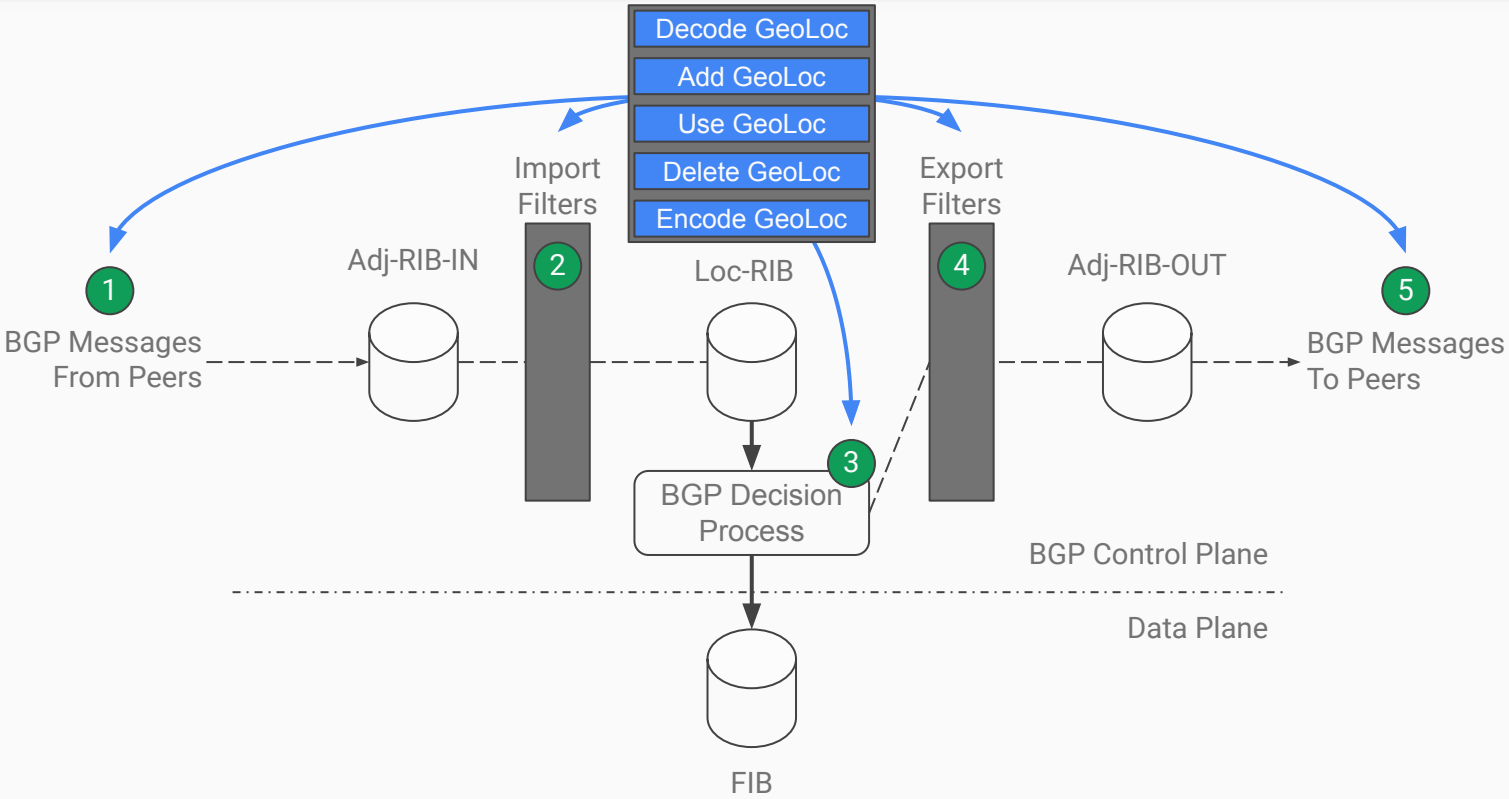
RFC 4271 BGP Workflow

GeoLoc needs to alter the BGP Workflow



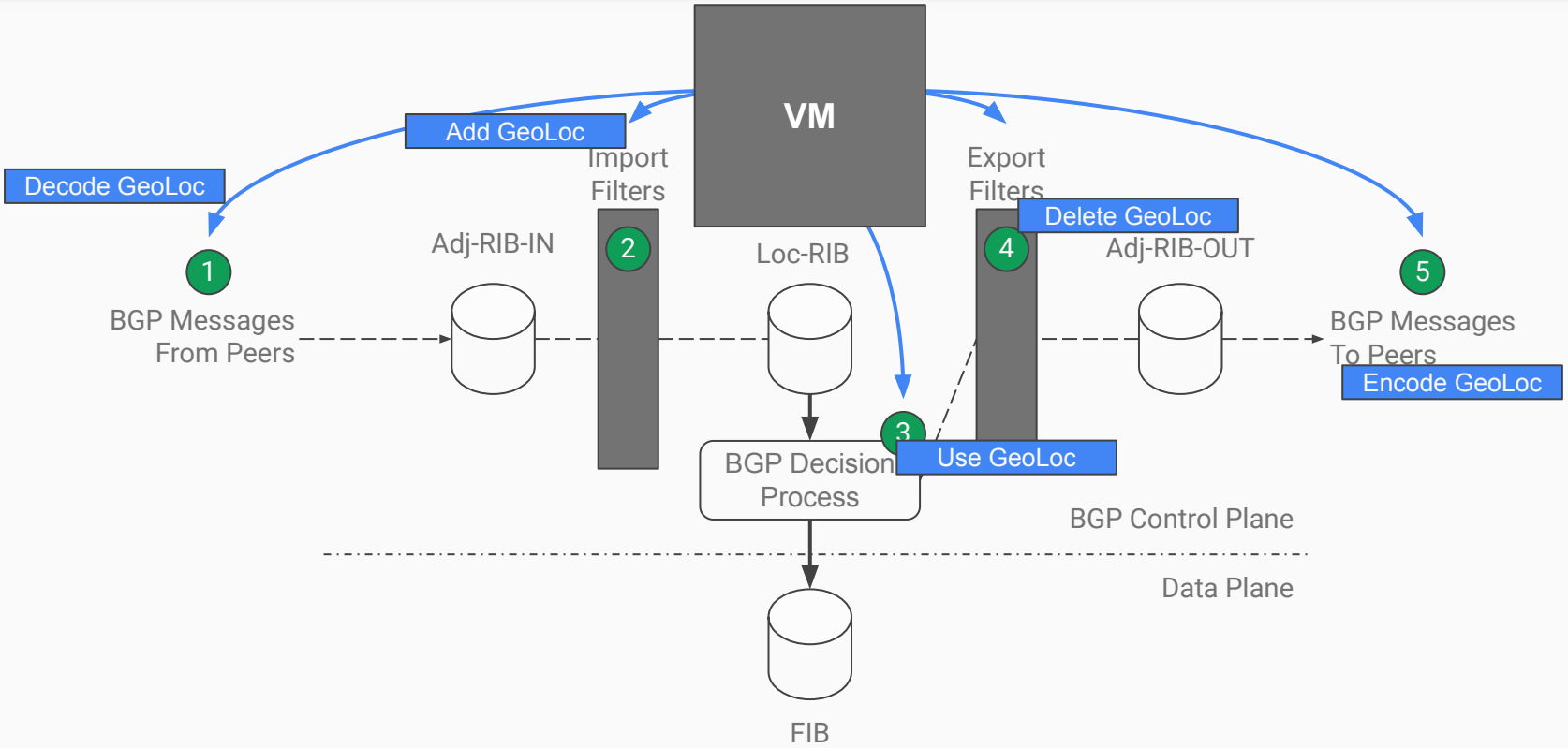
RFC 4271 BGP Workflow

GeoLoc needs to alter the BGP Workflow



RFC 4271 BGP Workflow

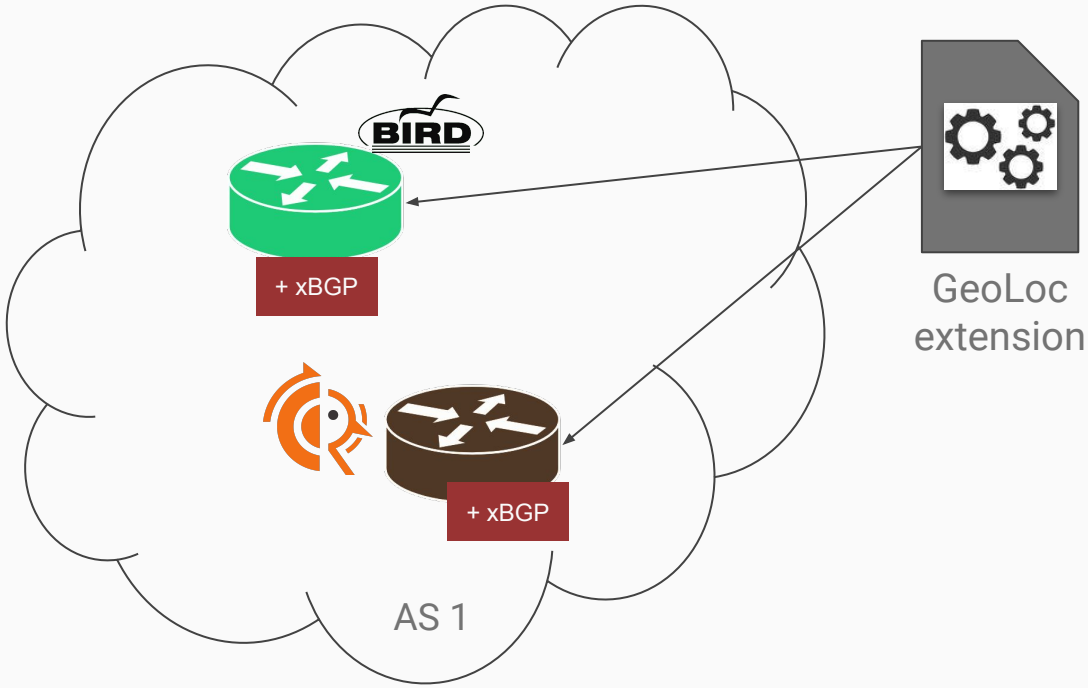
GeoLoc needs to alter the BGP Workflow



RFC 4271 BGP Workflow

xBGP: a paradigm shift

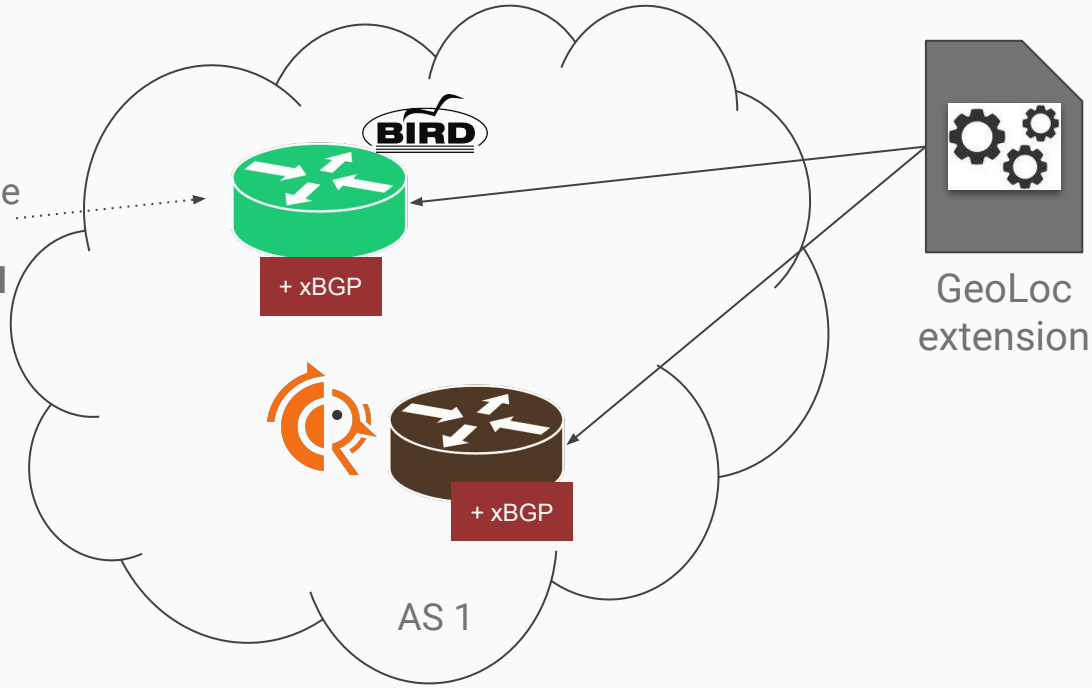
Operators can now add extension codes to their routers



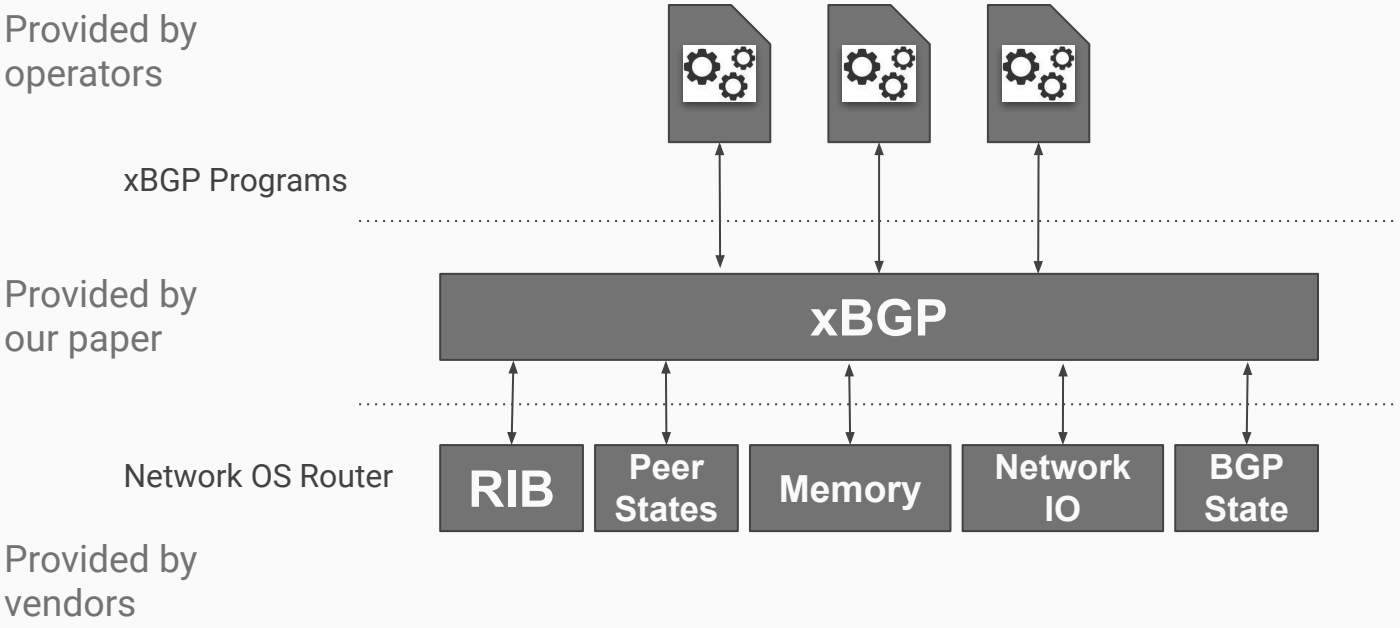
xBGP: a paradigm shift

Operators can now add extension codes to their routers

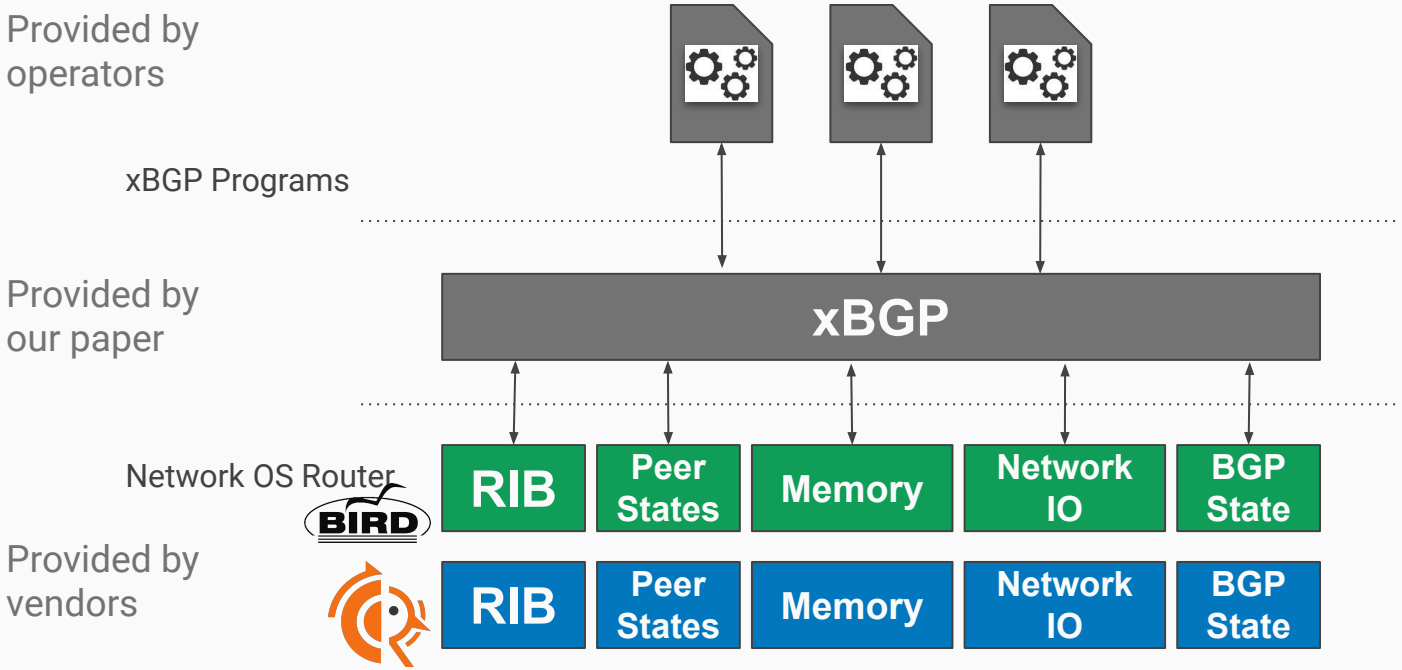
Routers execute the GeoLoc code with an **eBPF based VM**



xBGP makes the link between Router and extensions



xBGP makes the link between Router and extensions



Demonstrating the programmability of xBGP

xBGP requires a little adaptation to the host BGP implementation.

We have adapted both FRRouting and BIRD to be xBGP compliant

	FRRouting (LoC)	BIRD Routing (LoC)
Modification to the codebase	30	10
Building Insertion Points	73	66
Plugin API	624	415
`libxbgp`	3004 + dependencies	
User Space eBPF VM	2776	

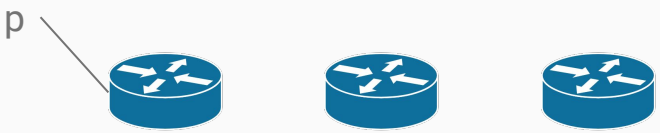
https://www.pluginized-protocols.org/xbgp

Other use cases

xBGP Extension	LoC
Geographical Location	388
Valley free routes	143
Filtering routes by IGP cost	36
Scanning for BGP zombies	1071
Influence remote BGP Decision Process	62
Monitoring the routes propagation time	806

⇒ Check the paper for those use cases

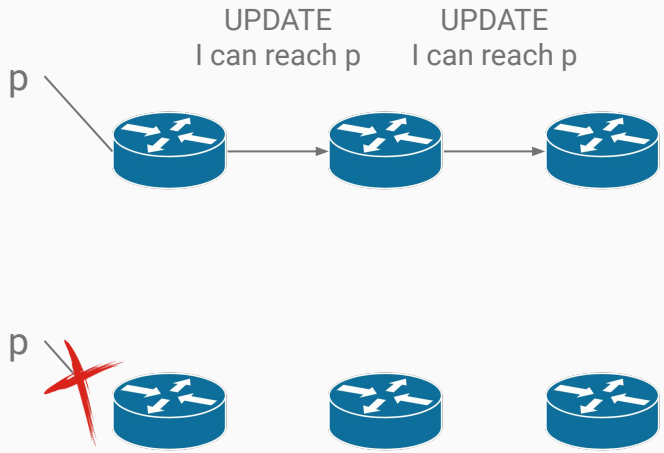
Detecting BGP Zombies



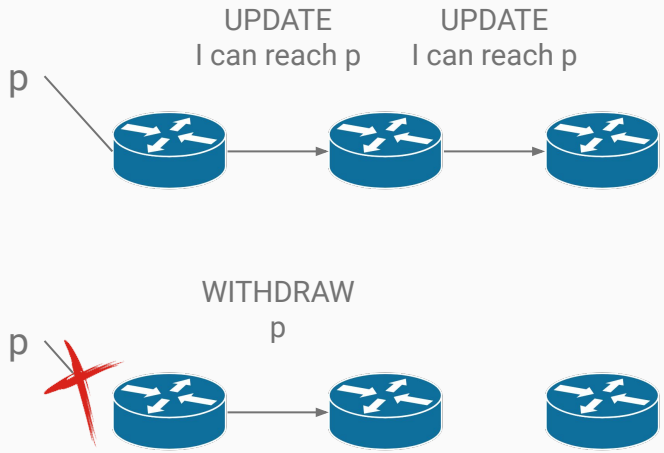
Detecting BGP Zombies

Detecting BGP Zombies

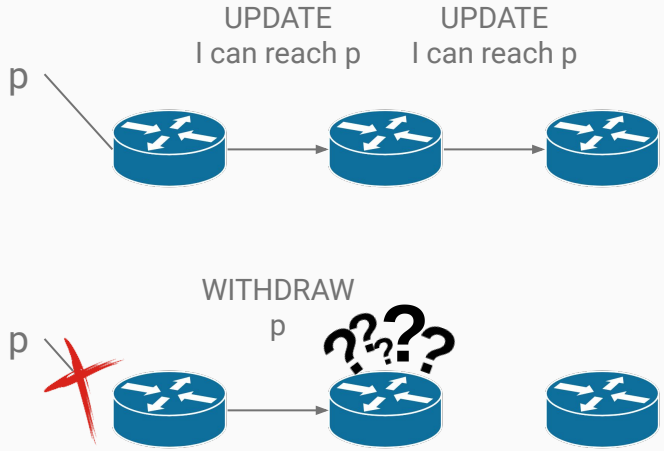
Detecting BGP Zombies



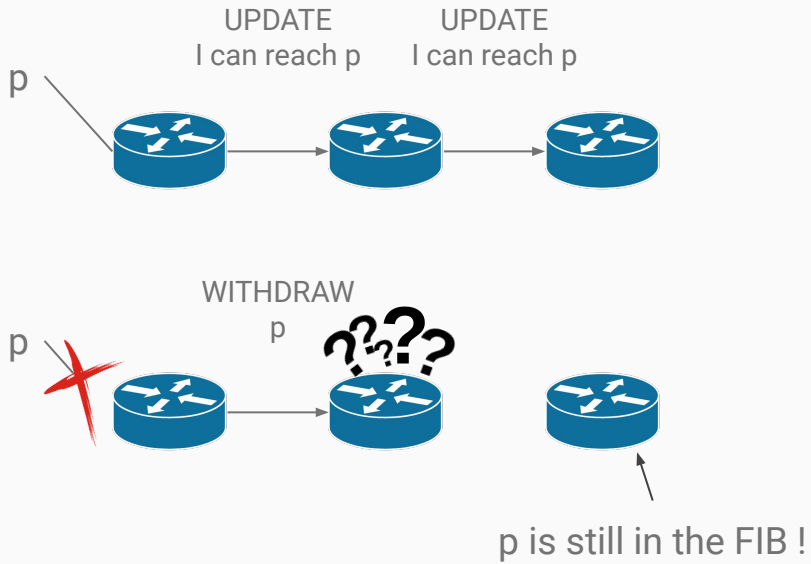
Detecting BGP Zombies



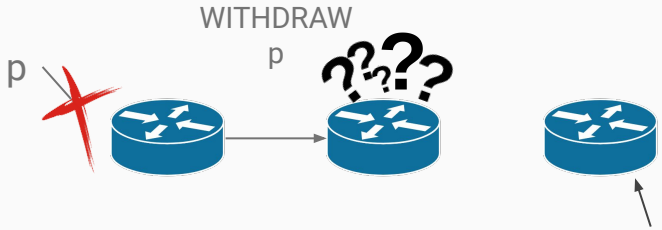
Detecting BGP Zombies



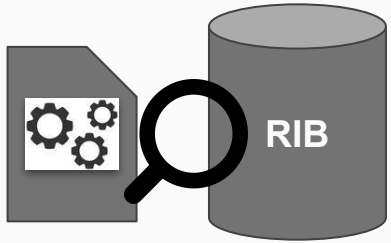
Detecting BGP Zombies



Detecting BGP Zombies

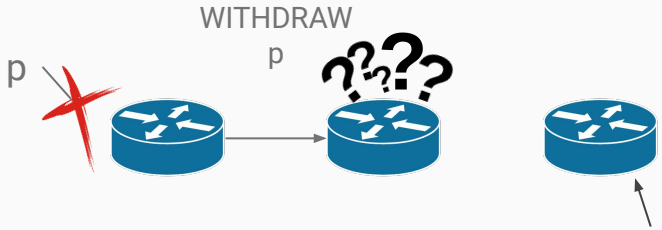


p is still in the FIB !

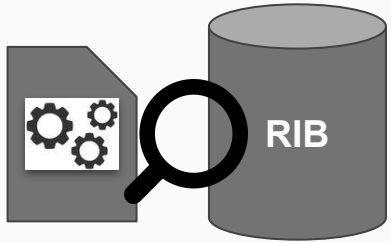


Checks routes older than
<x> <unit of time>

Detecting BGP Zombies



p is still in the FIB !

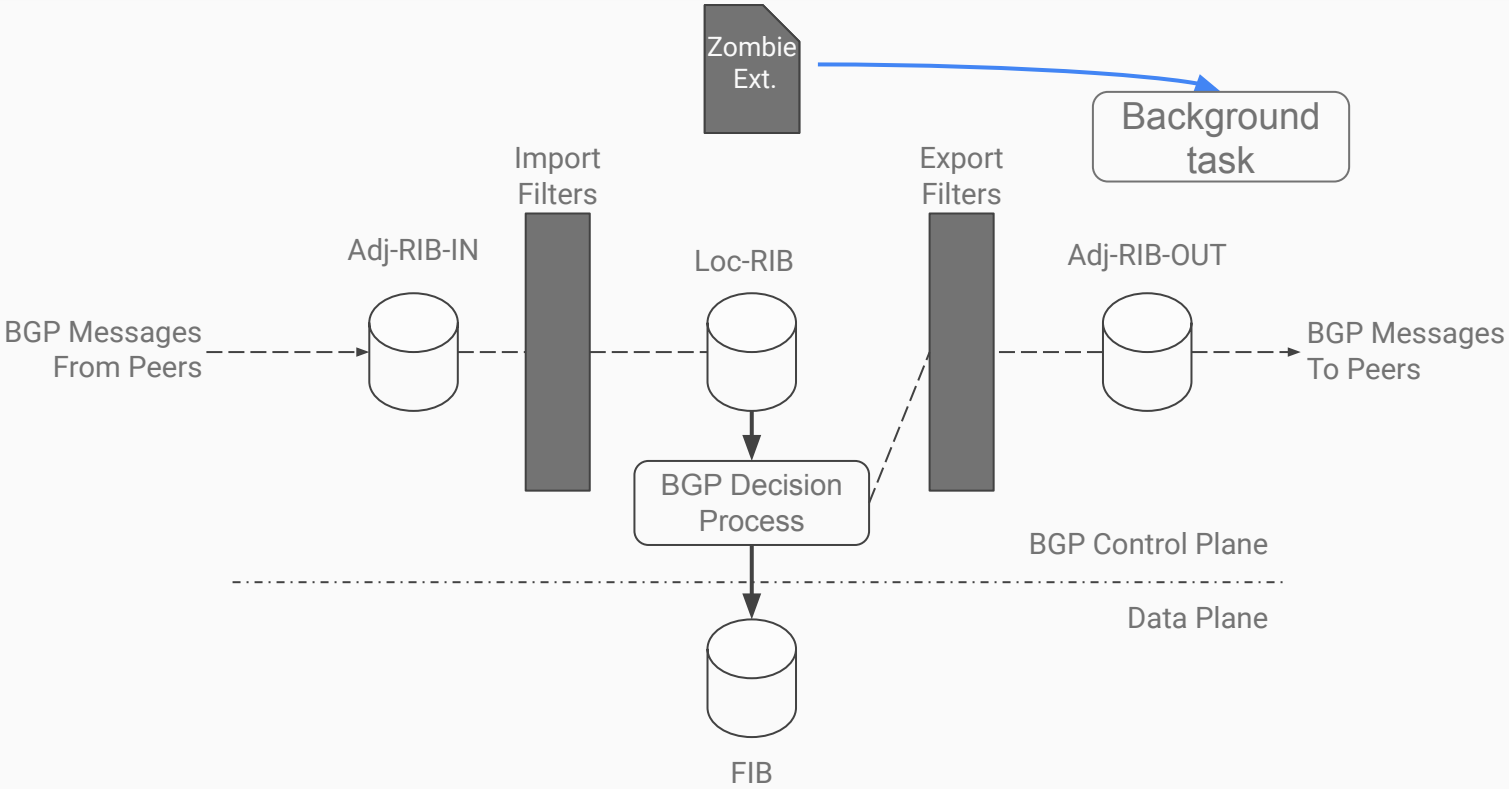


Checks routes older than
<x> <unit of time>

p since 4w 7h 36m 2s

Ask the upstream router to confirm if the route is still valid

Detecting BGP Zombies



RFC 4271 BGP Workflow

Agenda

- Why bringing programmability to BGP ?
- Inside xBGP
- **Does using xBGP have an impact on router performances ?**
- Verifying xBGP extensions
- Conclusion

Using a Virtual Machine inside BGP

Full IPv4 & IPv6 routing table (1M routes)

xBGP router

Downstream

Use Case	Convergence Time	
	xFRR	xBIRD

Using a Virtual Machine inside BGP

Full IPv4 & IPv6 routing table (1M routes)

xBGP router

Downstream

Additional overhead due to the xBGP internals

Use Case	Convergence Time	
	xFRR	xBIRD
No xBGP program	+1.05%	+1.60%

Using a Virtual Machine inside BGP

Full IPv4 & IPv6 routing table (1M routes)

xBGP router

Downstream

Use Case	Convergence Time	
	xFRR	xBIRD
No xBGP program	+1.05%	+1.60%
Route reflection	+12.97%	+7.43%

Additional overhead due to the xBGP internals

Worst case involving all insertion points

Using a Virtual Machine inside BGP

Full IPv4 & IPv6 routing table (1M routes)

xBGP router

Downstream

Additional overhead due to the xBGP internals

Worst case involving all insertion points

Use Case	Convergence Time	
	xFRR	xBIRD
No xBGP program	+1.05%	+1.60%
Route reflection	+12.97%	+7.43%

Data serialization is more costly in FRR

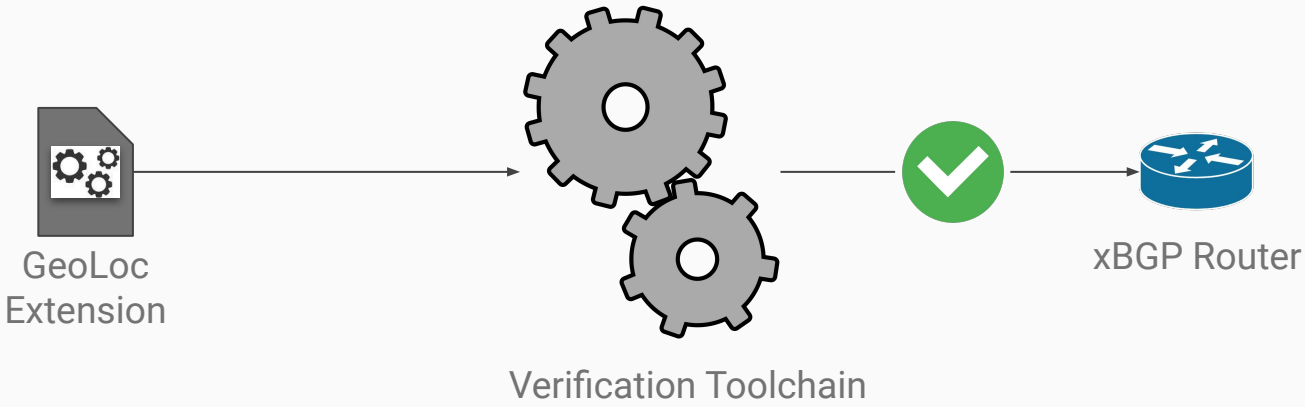
\+ The “JIT” compiler is not efficient as native machine code

Agenda

- Why bringing programmability to BGP ?
- Inside xBGP
- Does using xBGP have an impact on router performances ?
- **Verifying xBGP extensions**
- Conclusion

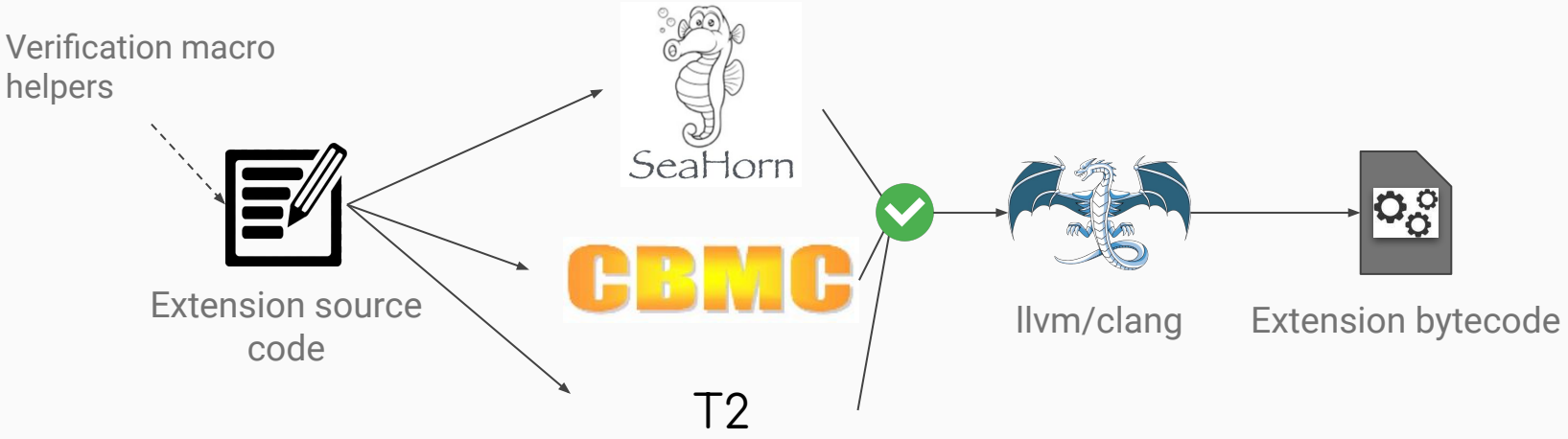
The code executed by xBGP is **untrusted**

Could the GeoLoc extension break BGP ?



The code executed by xBGP is **untrusted** (cont.)

The code should be annotated, and then passed to the verification tools.



Offline verification tools

The right tool to the right property

- **T2**: termination
- **CBMC**: memory safety
- **libxbgp**: VM isolation & API restriction

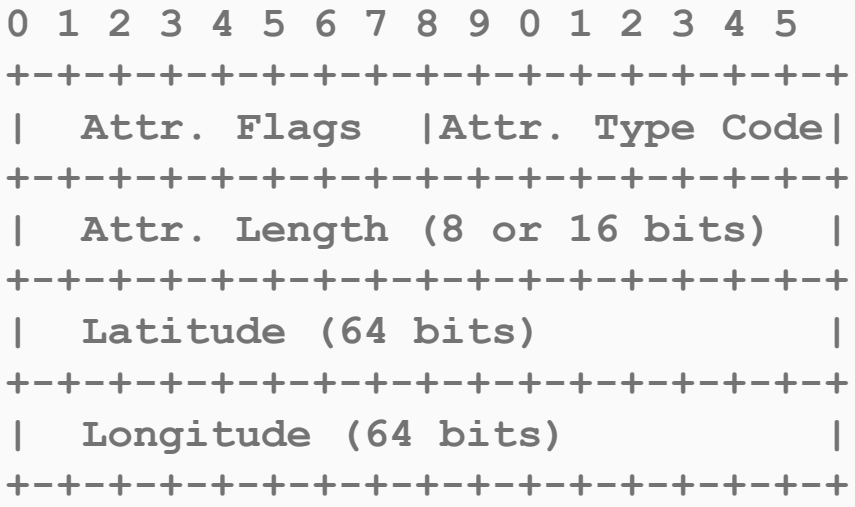
Basic properties

- **Seahorn**: BGP properties

Properties related to BGP

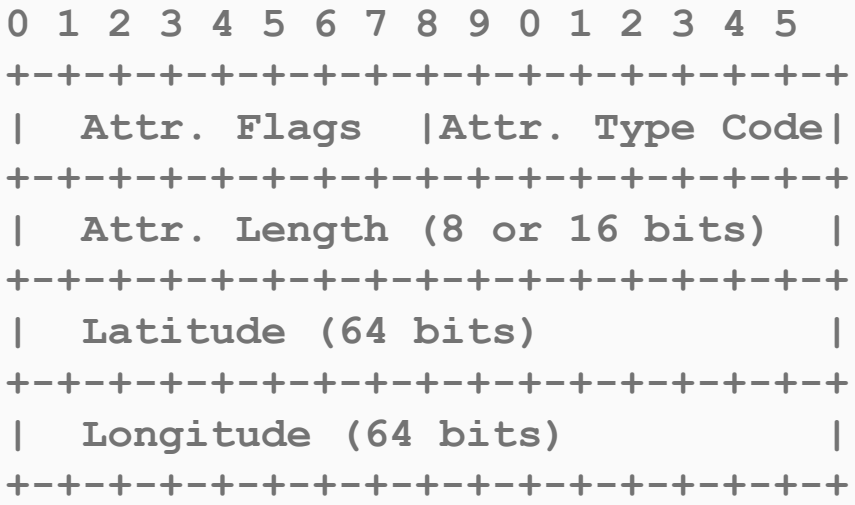
Verifying the BGP syntax of GeoLoc

If the xBGP extension adds Geographic coordinates, it must respect the TLV format defined in the draft.



Verifying the BGP syntax of GeoLoc

If the xBGP extension adds Geographic coordinates, it must respect the TLV format defined in the draft.



Flags **MUST** be 0x8

Code **MUST** be 0x2a

Length **MUST** be 8

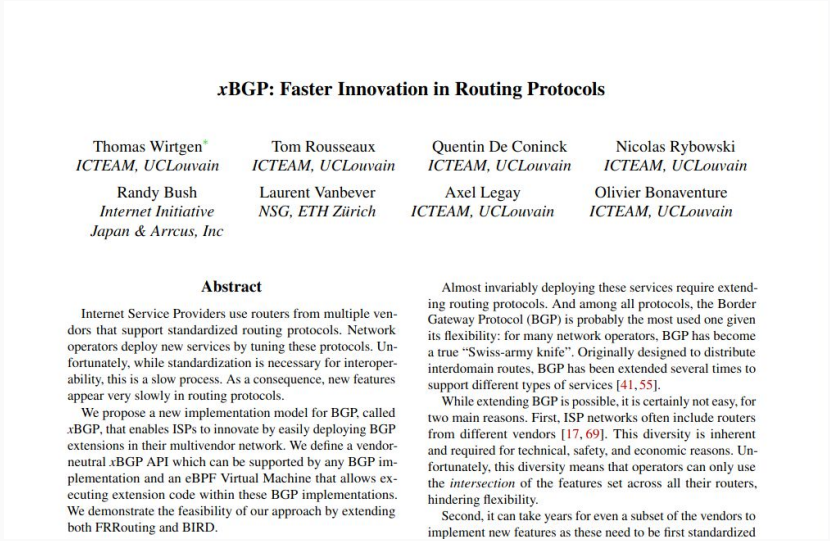
<Latitude, Longitude> **MUST** be a valid geographic location

Conclusion

With xBGP, BGP implementations can become truly extensible

See https://www.pluginized-protocols.org/xbgp for running source code

xBGP provides new opportunities with other routing protocols



xBGP: Faster Innovation in Routing Protocols

Thomas Wirtgen* *ICTEAM, UCLouvain*
Tom Rousseaux *ICTEAM, UCLouvain*
Quentin De Coninck *ICTEAM, UCLouvain*
Nicolas Rybowski *ICTEAM, UCLouvain*
Randy Bush *Internet Initiative Japan & Arrcus, Inc*
Laurent Vanbever *NSG, ETH Zürich*
Axel Legay *ICTEAM, UCLouvain*
Olivier Bonaventure *ICTEAM, UCLouvain*

Abstract

Internet Service Providers use routers from multiple vendors that support standardized routing protocols. Network operators deploy new services by tuning these protocols. Unfortunately, while standardization is necessary for interoperability, this is a slow process. As a consequence, new features appear very slowly in routing protocols.

We propose a new implementation model for BGP, called xBGP, that enables ISPs to innovate by easily deploying BGP extensions in their multivendor network. We define a vendor-neutral xBGP API which can be supported by any BGP implementation and an eBPF Virtual Machine that allows executing extension code within these BGP implementations. We demonstrate the feasibility of our approach by extending both FRRouting and BIRD.

Almost invariably deploying these services require extending routing protocols. And among all protocols, the Border Gateway Protocol (BGP) is probably the most used one given its flexibility: for many network operators, BGP has become a true "Swiss-army knife". Originally designed to distribute interdomain routes, BGP has been extended several times to support different types of services [41, 55].

While extending BGP is possible, it is certainly not easy, for two main reasons. First, ISP networks often include routers from different vendors [17, 69]. This diversity is inherent and required for technical, safety, and economic reasons. Unfortunately, this diversity means that operators can only use the *intersection* of the features set across all their routers, hindering flexibility.

Second, it can take years for even a subset of the vendors to implement new features as these need to be first standardized

thomas.wirtgen@uclouvain.be