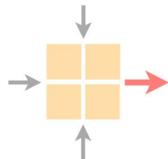


xBGP: Faster Innovation in Routing Protocols

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



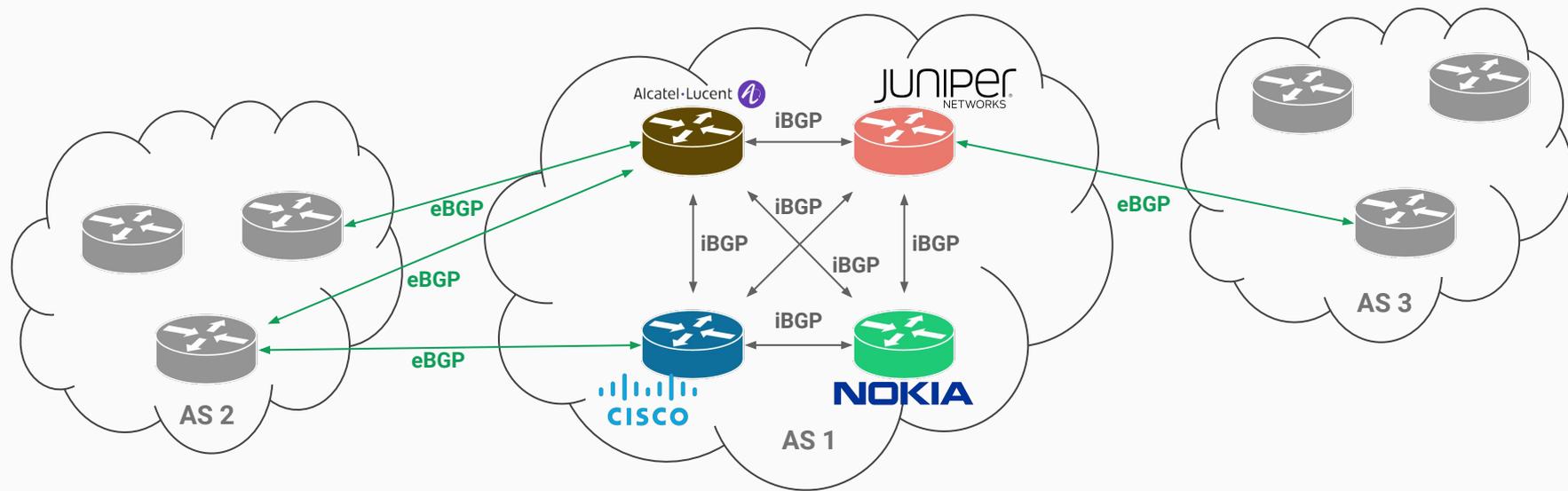
Networked Systems
ETH Zürich — seit 2015



Agenda

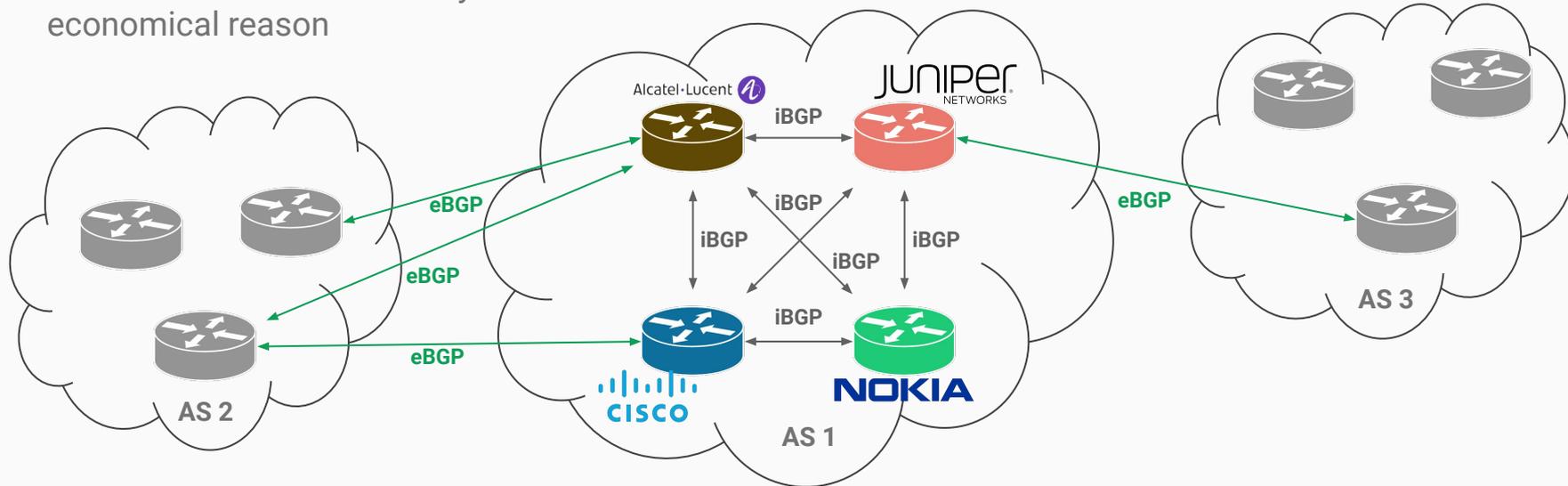
- **Why bringing programmability to BGP?**
- Inside xBGP
- Does using xBGP have an impact on router performances?
- Verifying xBGP extensions
- Conclusion

Routing on the Internet



Routing on the Internet

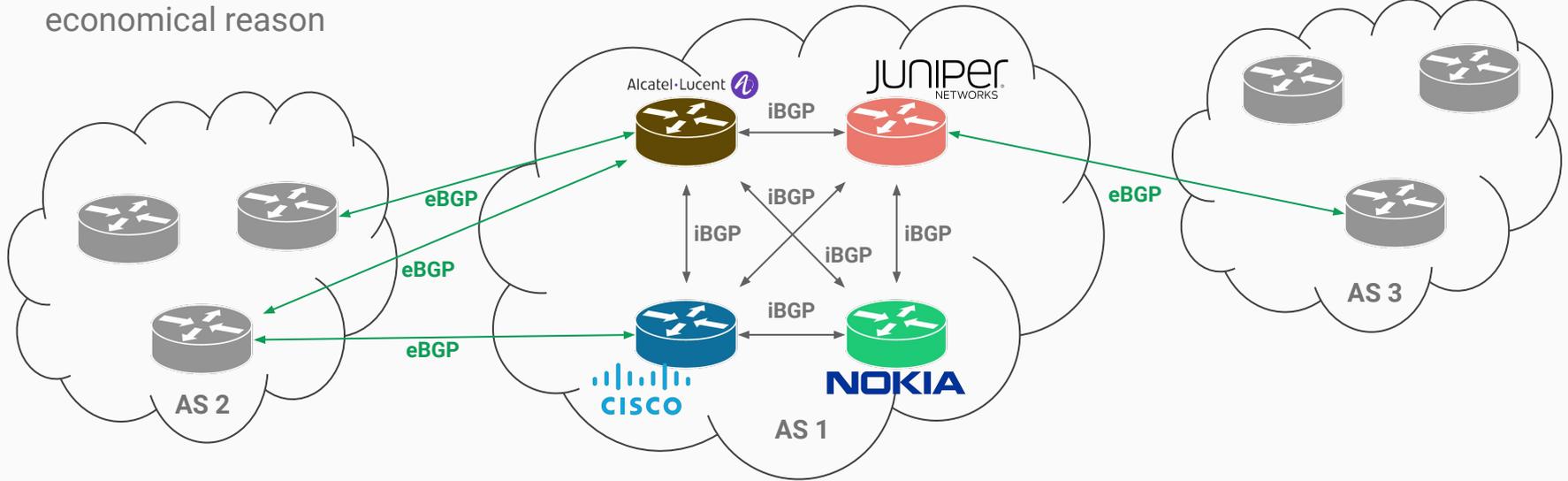
It's a best practice to have routers from different vendors for stability & economical reason



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

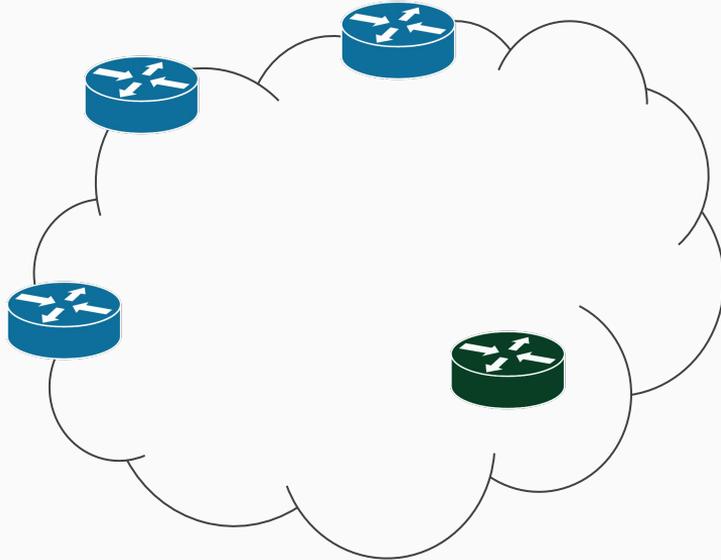
But they are limited:

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



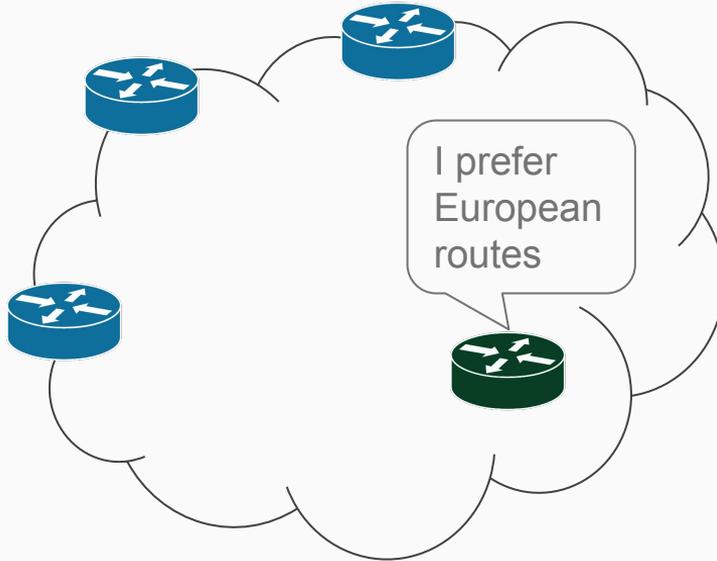
Enhancing the visibility of the BGP control plane

Intra domain routers have no information about the exit router



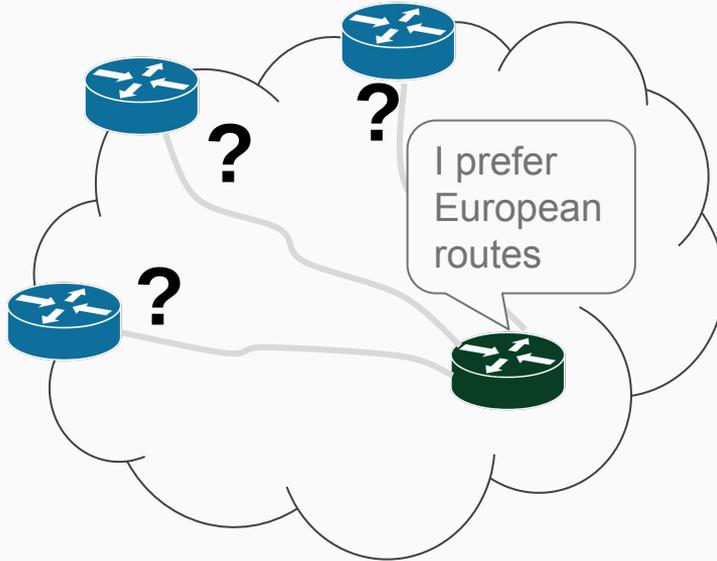
Enhancing the visibility of the BGP control plane

Intra domain routers have no information about the exit router



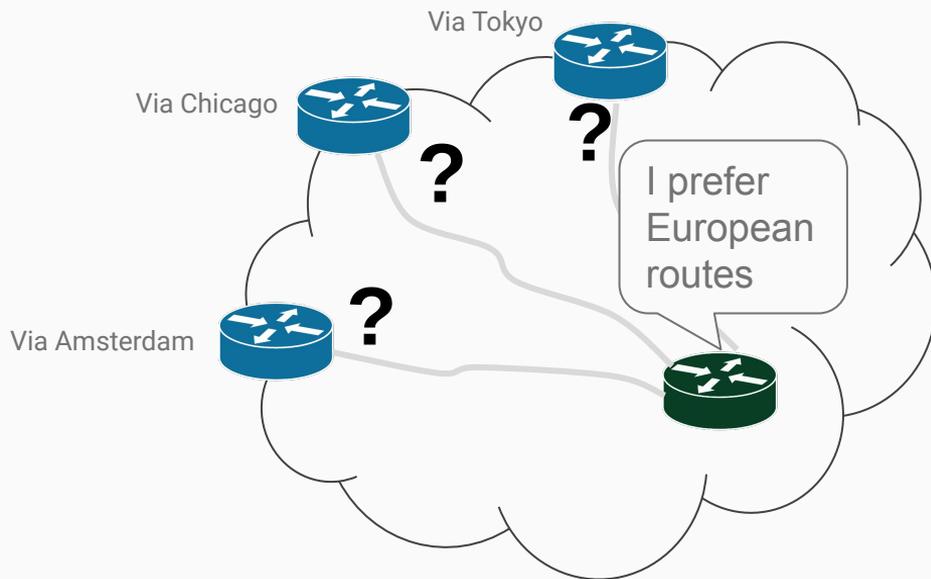
Enhancing the visibility of the BGP control plane

Intra domain routers have no information about the exit router



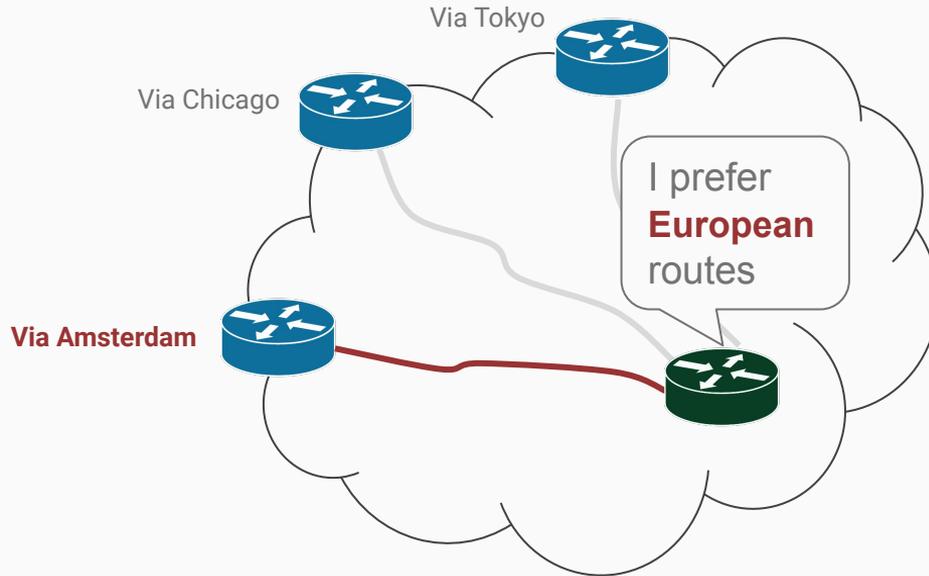
Enhancing the visibility of the BGP control plane

Intra domain routers have no information about the exit router



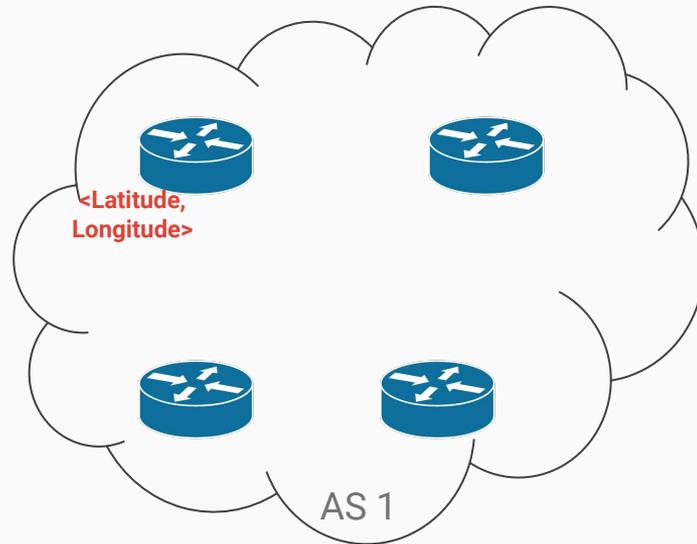
Enhancing the visibility of the BGP control plane

Intra domain routers have no information about the exit router



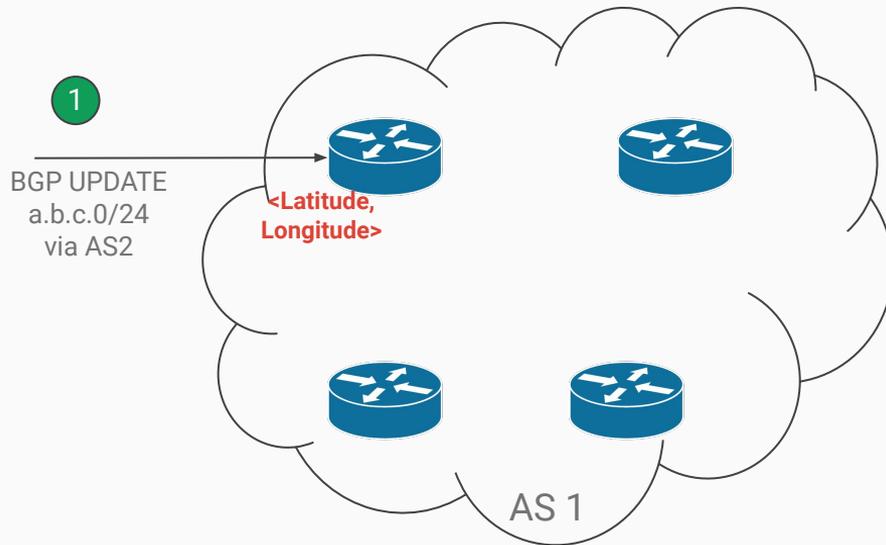
A complex feature to achieve with classical routers

The Geographical Location TLV (GeoLoc TLV)



A complex feature to achieve with classical routers

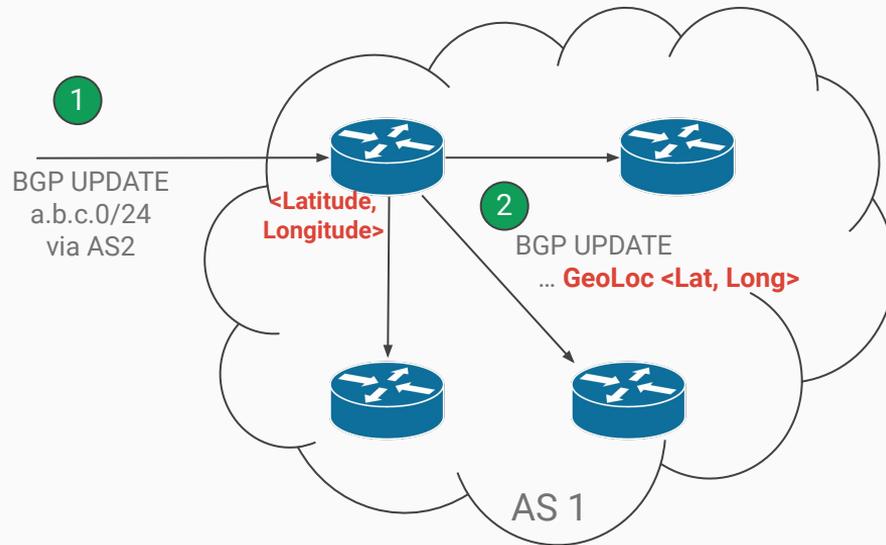
The Geographical Location TLV (GeoLoc TLV)



- 1 Add GeoLoc on the input edge routers

A complex feature to achieve with classical routers

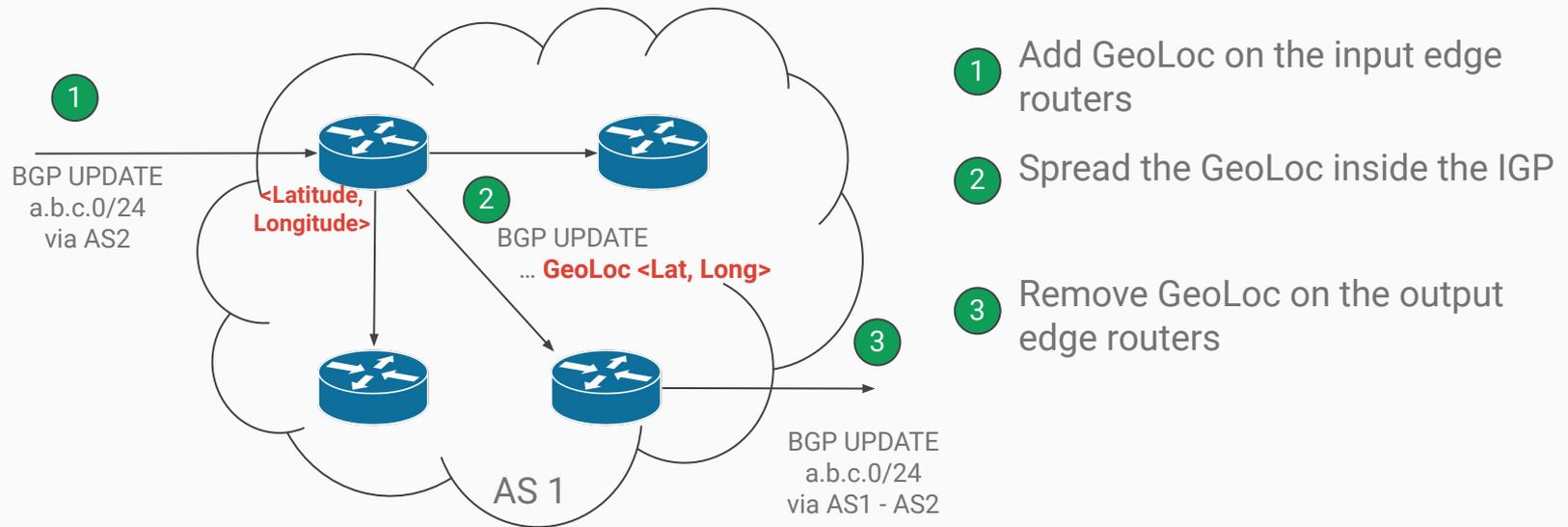
The Geographical Location TLV (GeoLoc TLV)



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

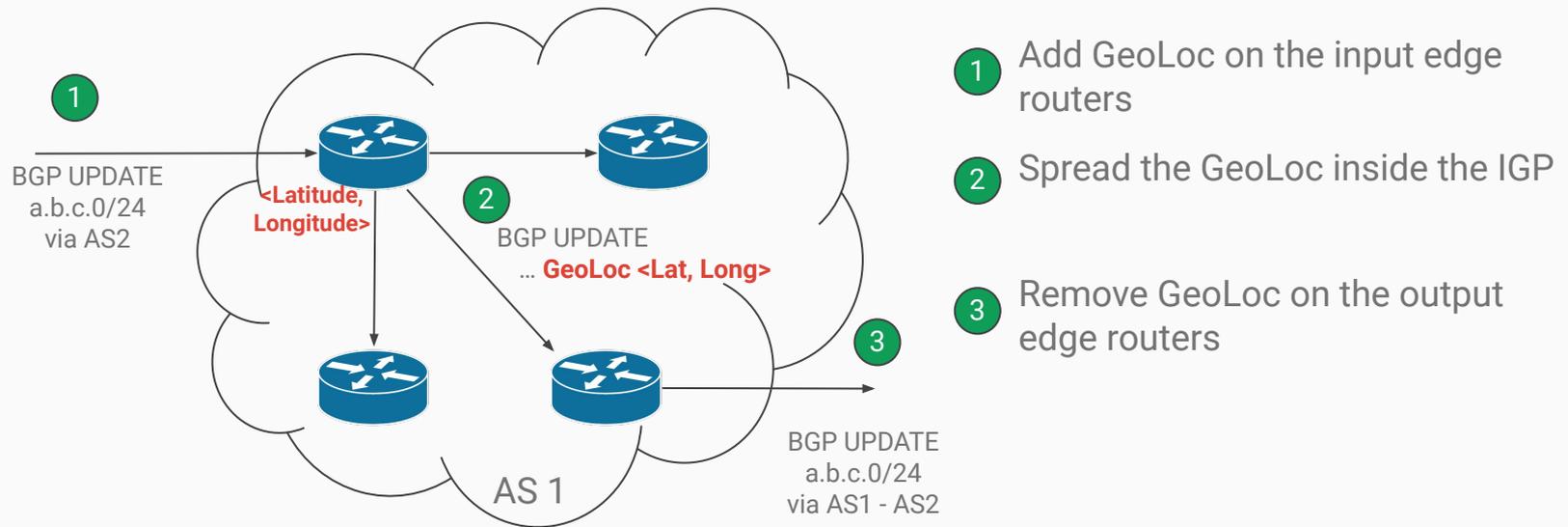
A complex feature to achieve with classical routers

The Geographical Location TLV (GeoLoc TLV)



A complex feature to achieve with classical routers

The Geographical Location TLV (GeoLoc TLV)

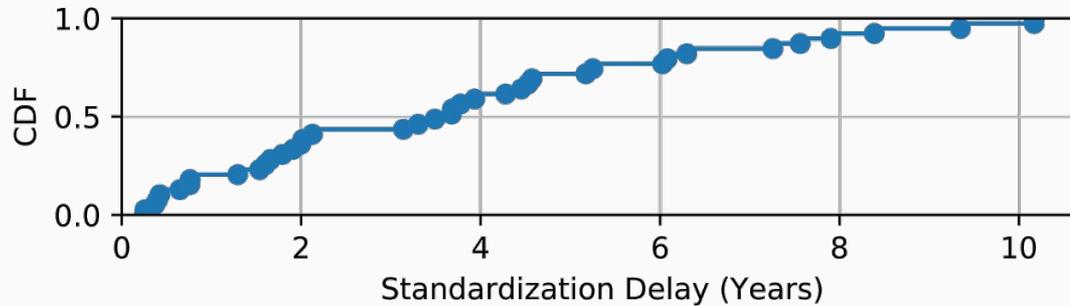


⇒ This requires to have access to the router's BGP implementation

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



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!



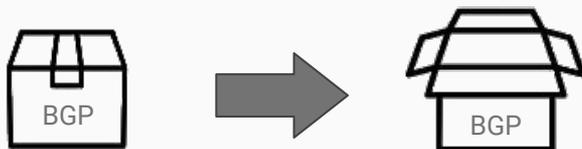
Current paradigm slows innovation

Problem #1: Routers from different vendors

Problem #2: Protocol extensions not implemented on all routers

Problem #3: Slow upgrade process

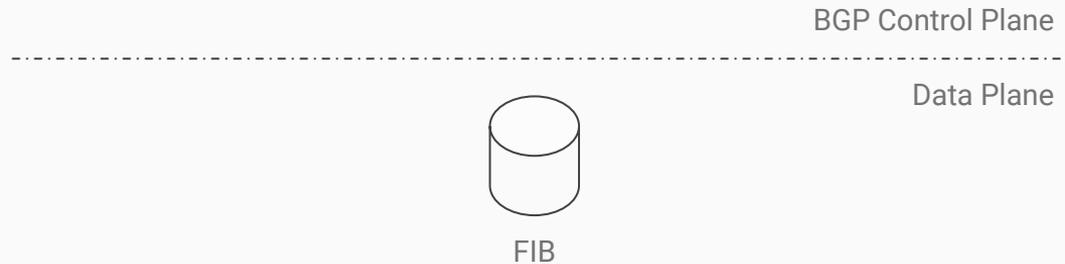
⇒ xBGP is designed to bring **innovation & programmability** to existing routing protocols



Agenda

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

GeoLoc needs to alter the BGP Workflow



GeoLoc needs to alter the BGP Workflow

BGP Messages
From Peers

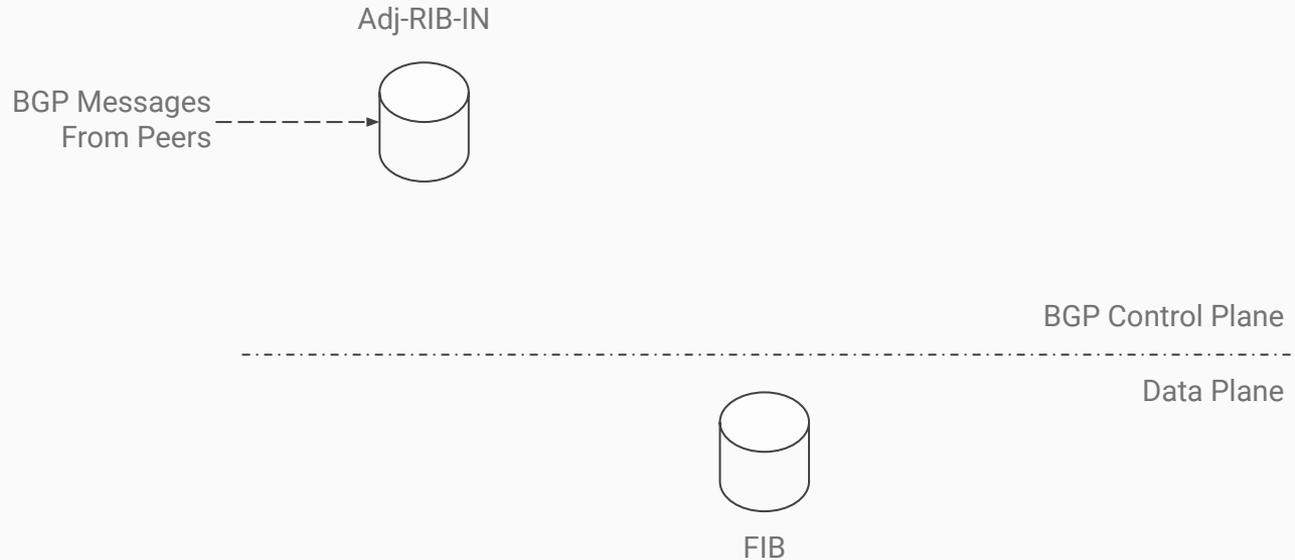
BGP Control Plane

Data Plane

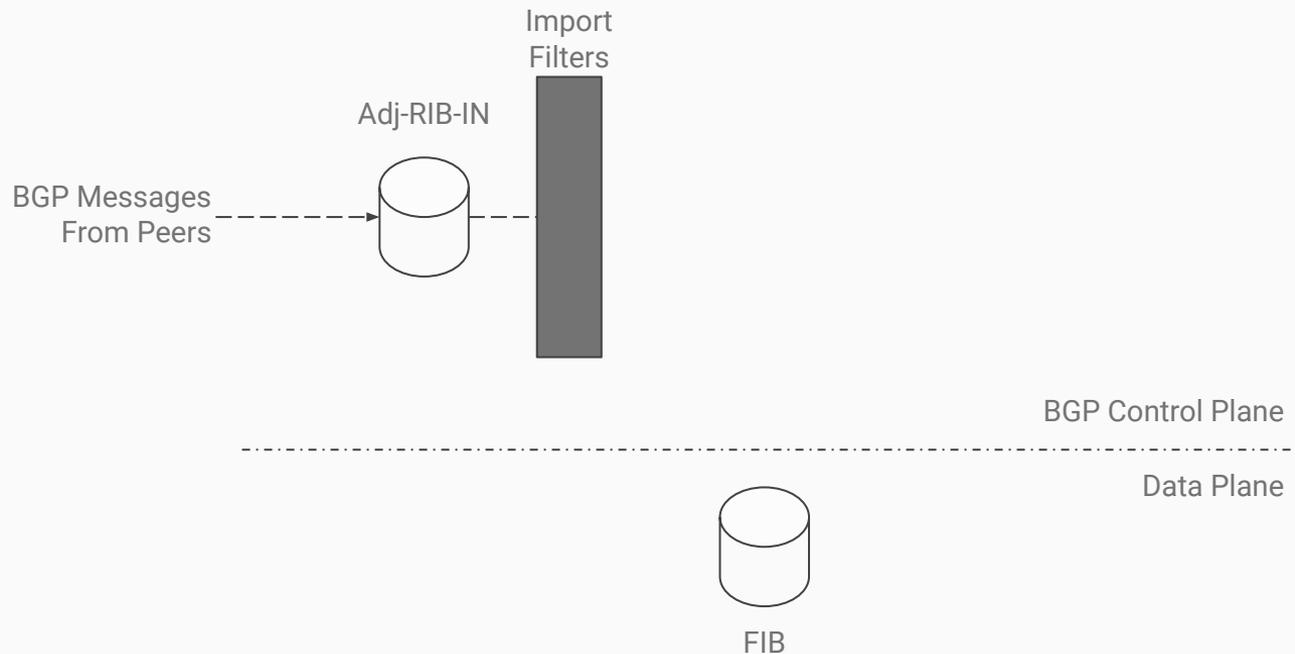


FIB

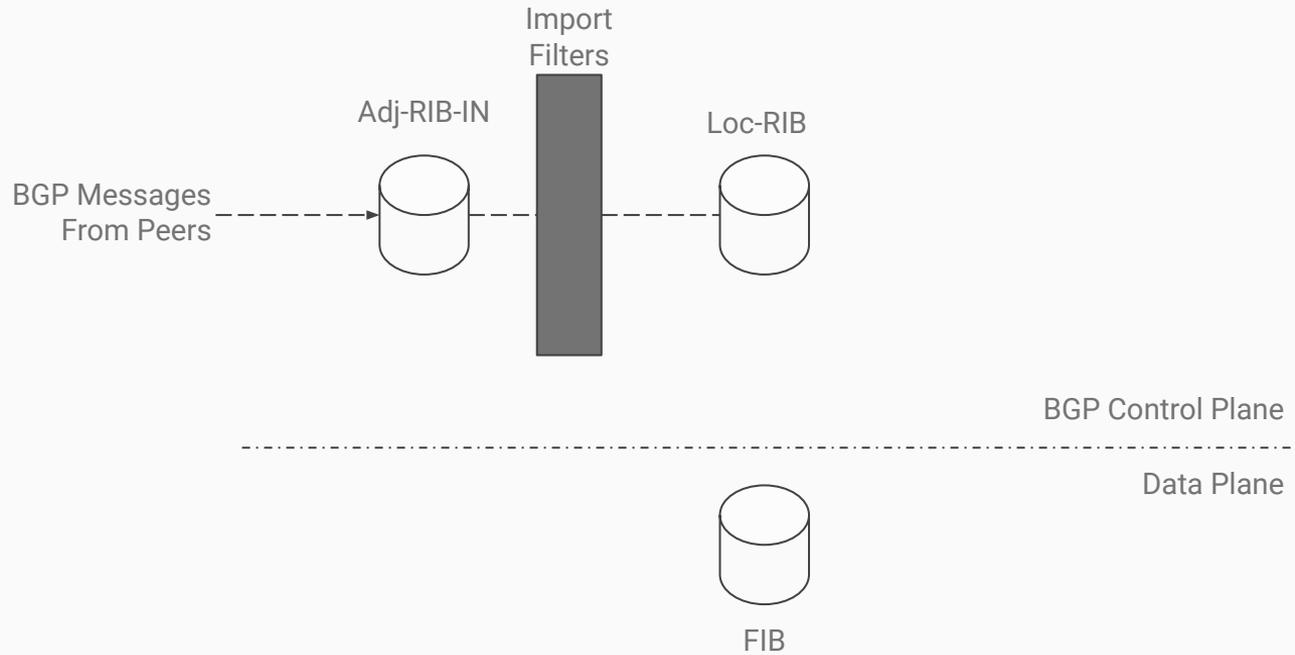
GeoLoc needs to alter the BGP Workflow



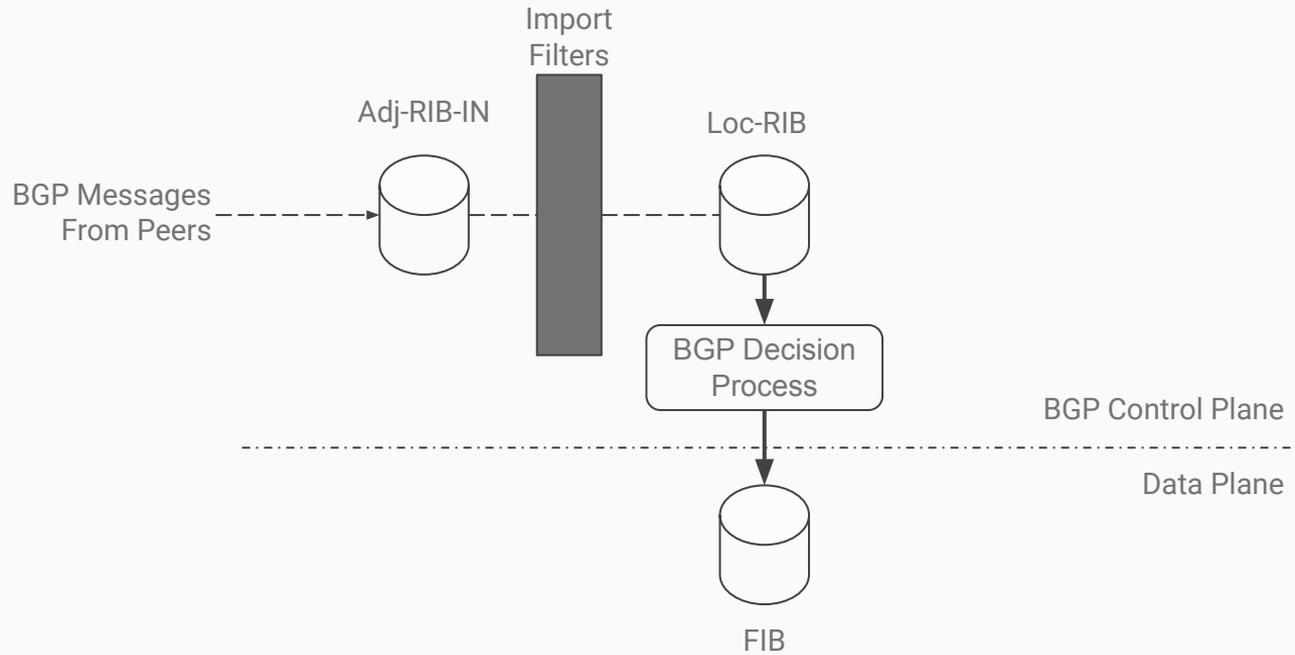
GeoLoc needs to alter the BGP Workflow



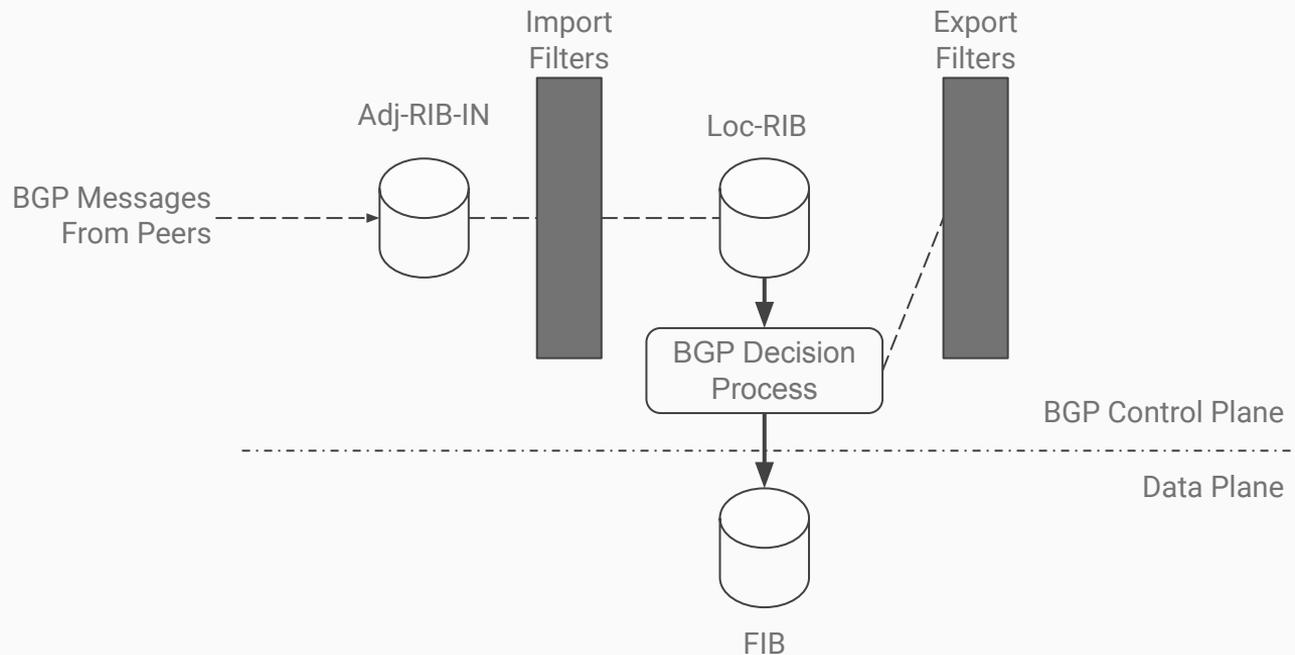
GeoLoc needs to alter the BGP Workflow



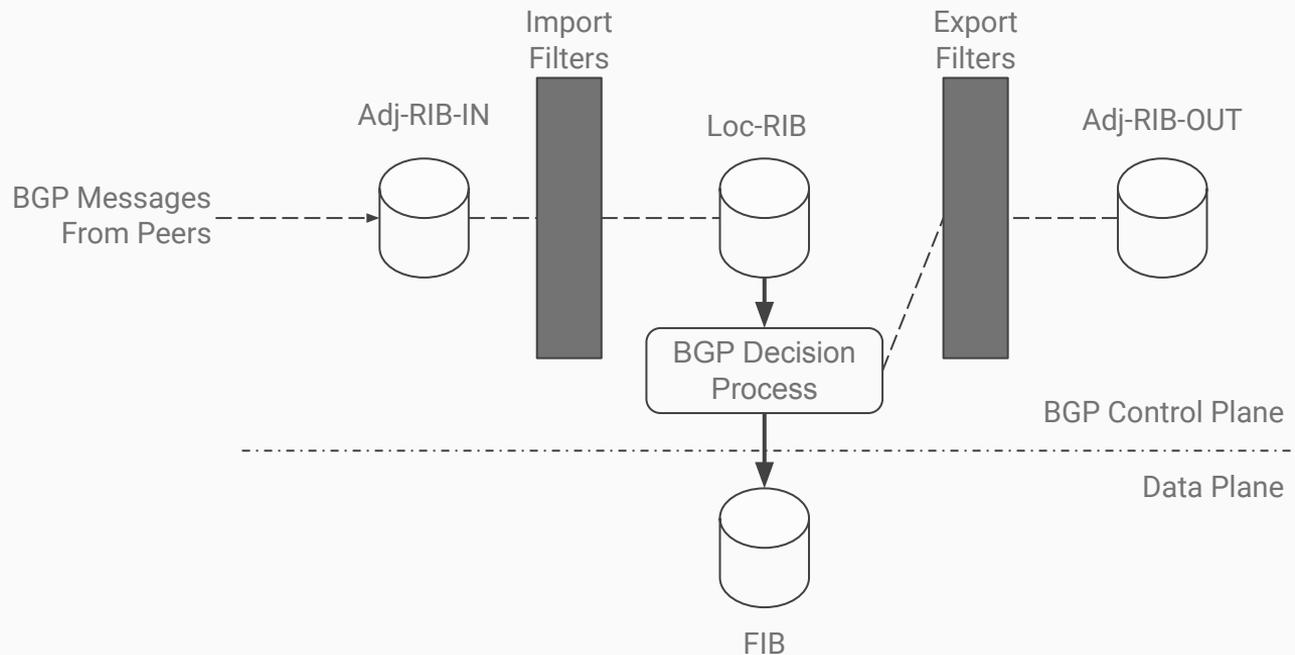
GeoLoc needs to alter the BGP Workflow



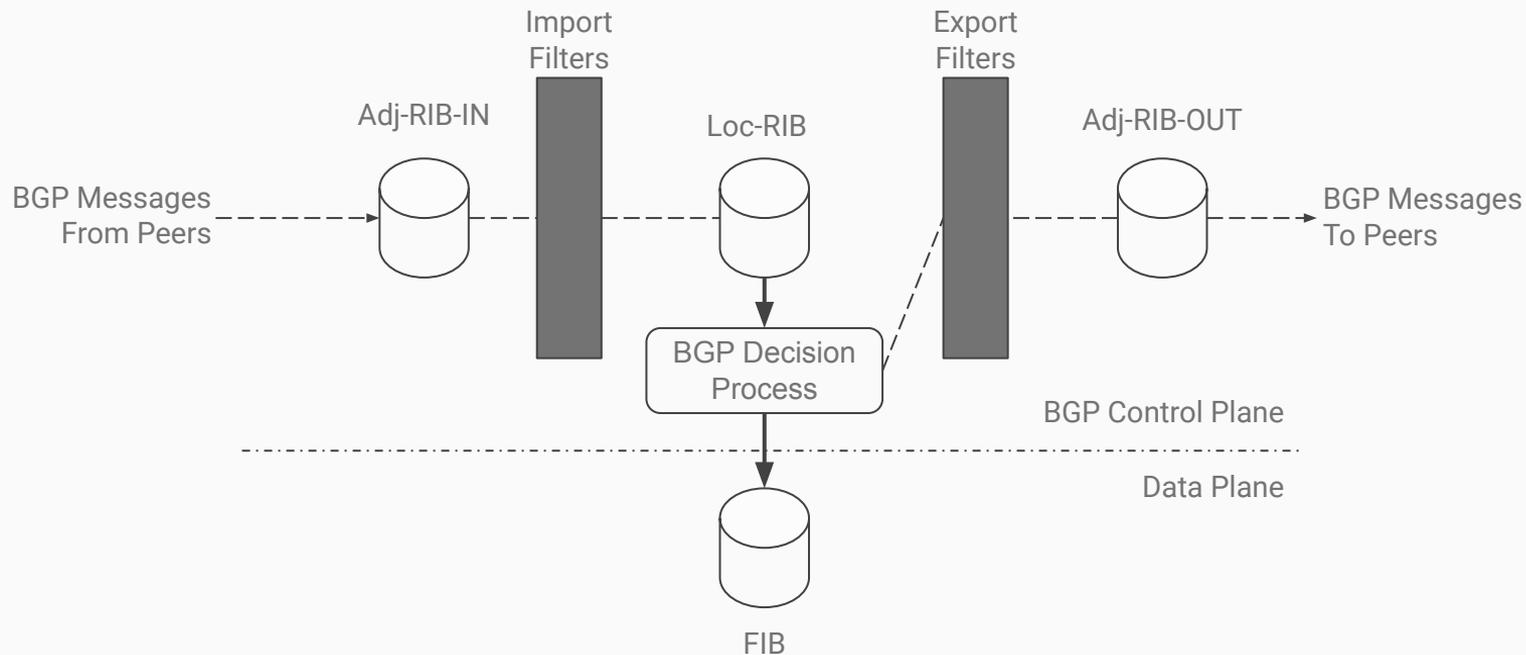
GeoLoc needs to alter the BGP Workflow



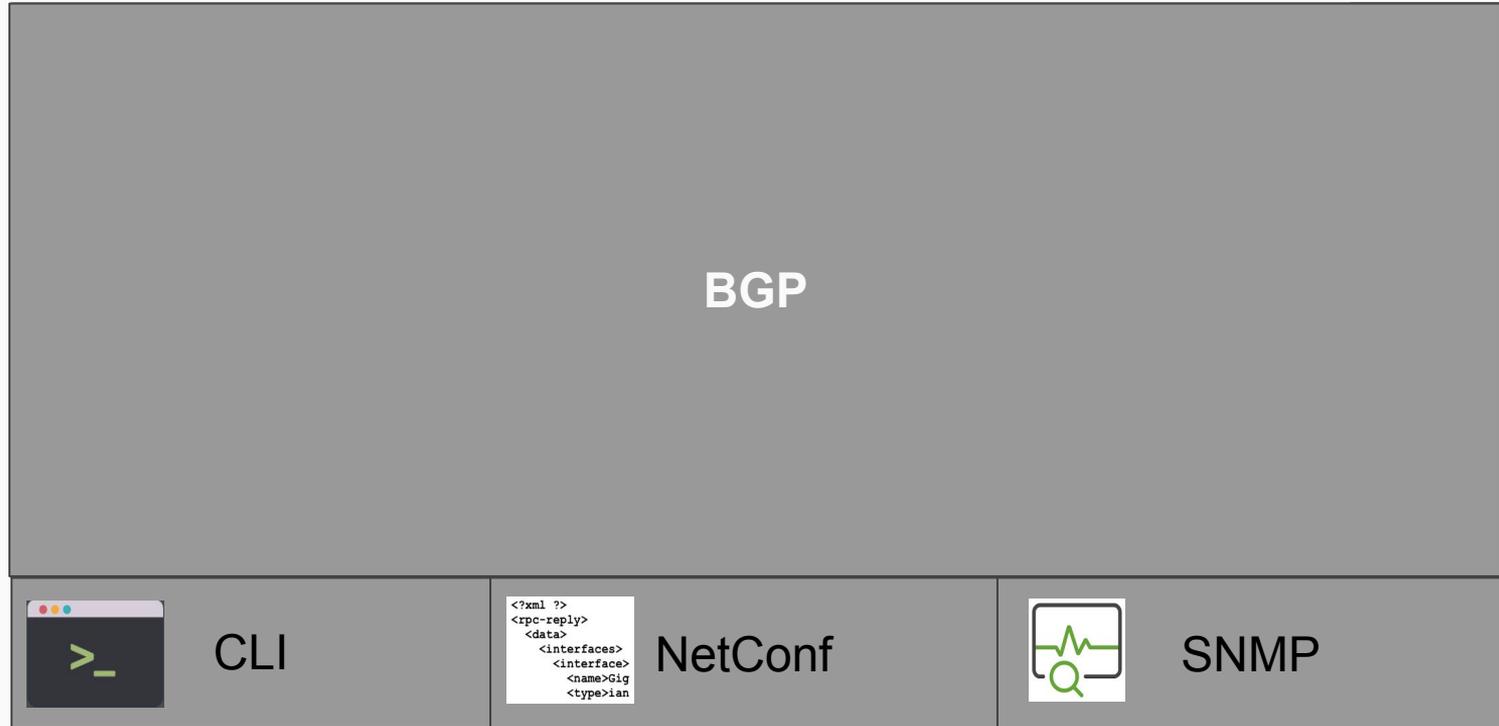
GeoLoc needs to alter the BGP Workflow



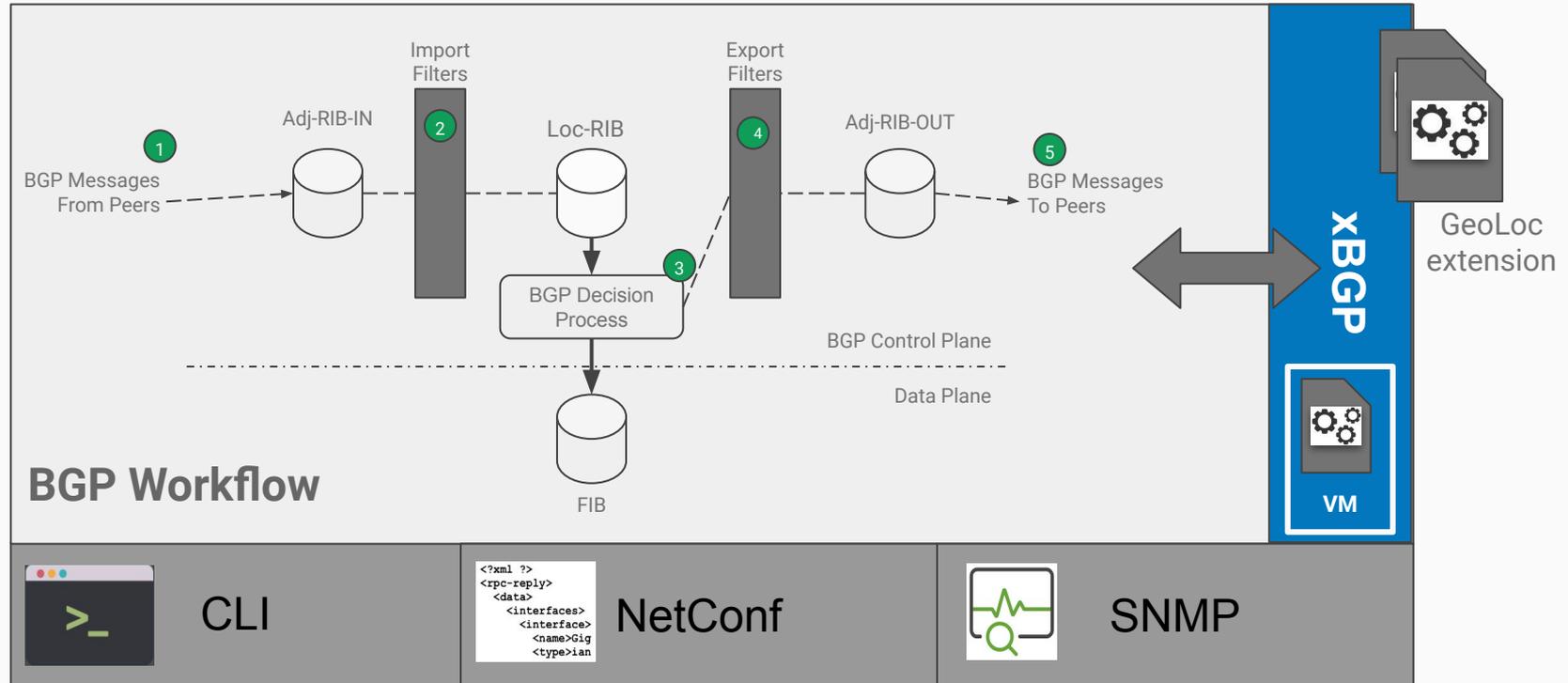
GeoLoc needs to alter the BGP Workflow



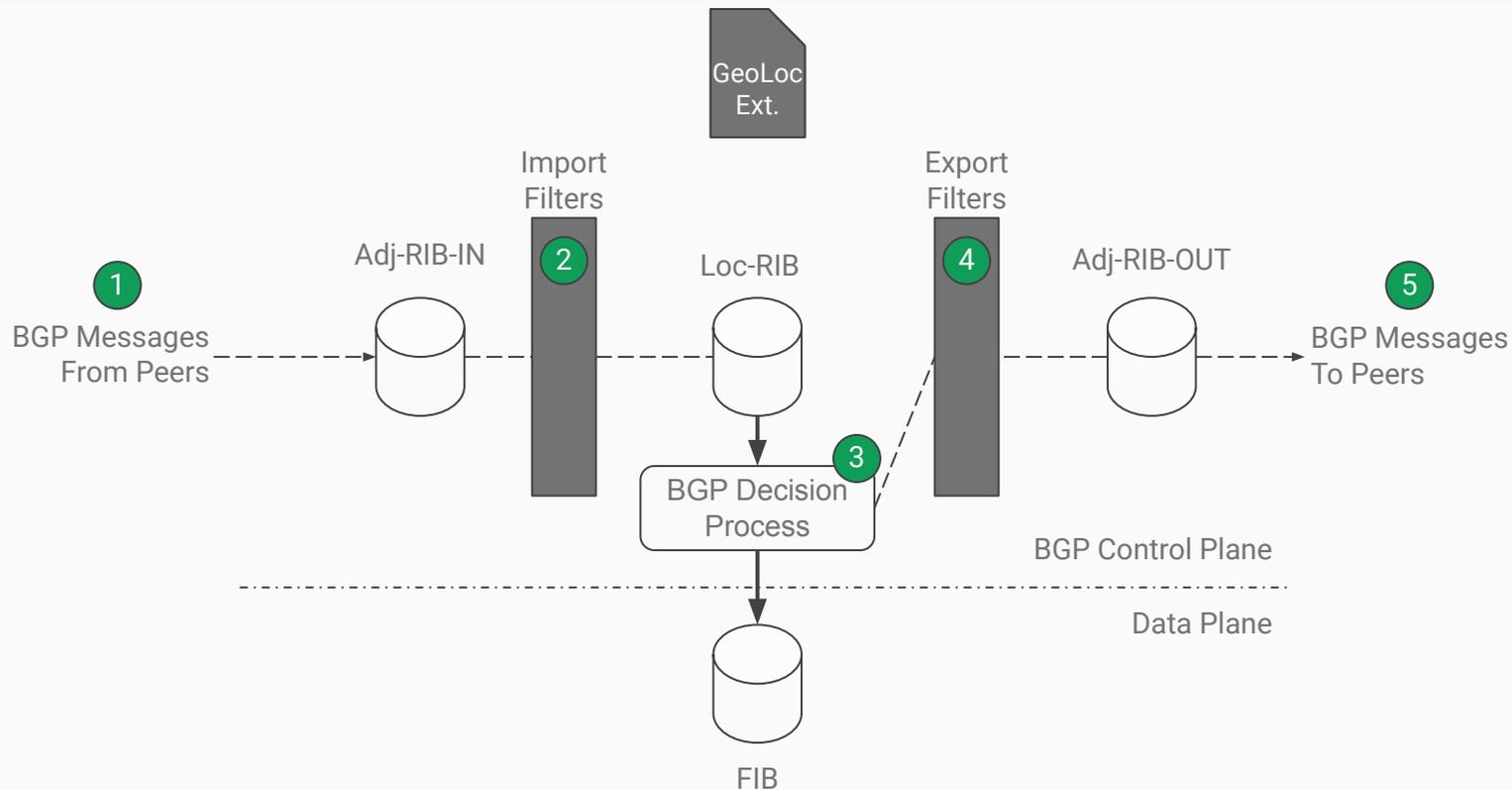
Traditional BGP implementations are opaque



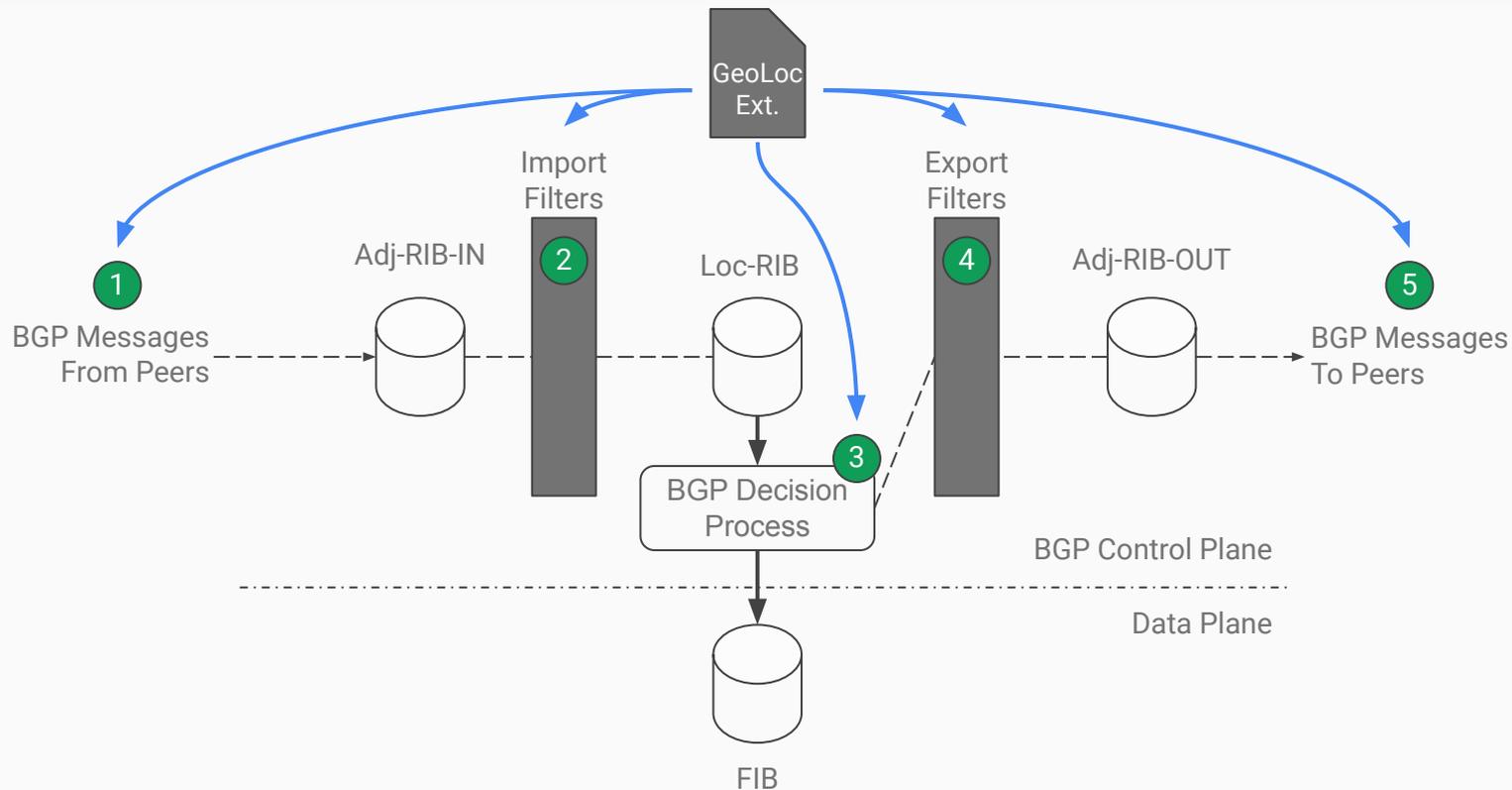
BGP workflow are now exposed with xBGP



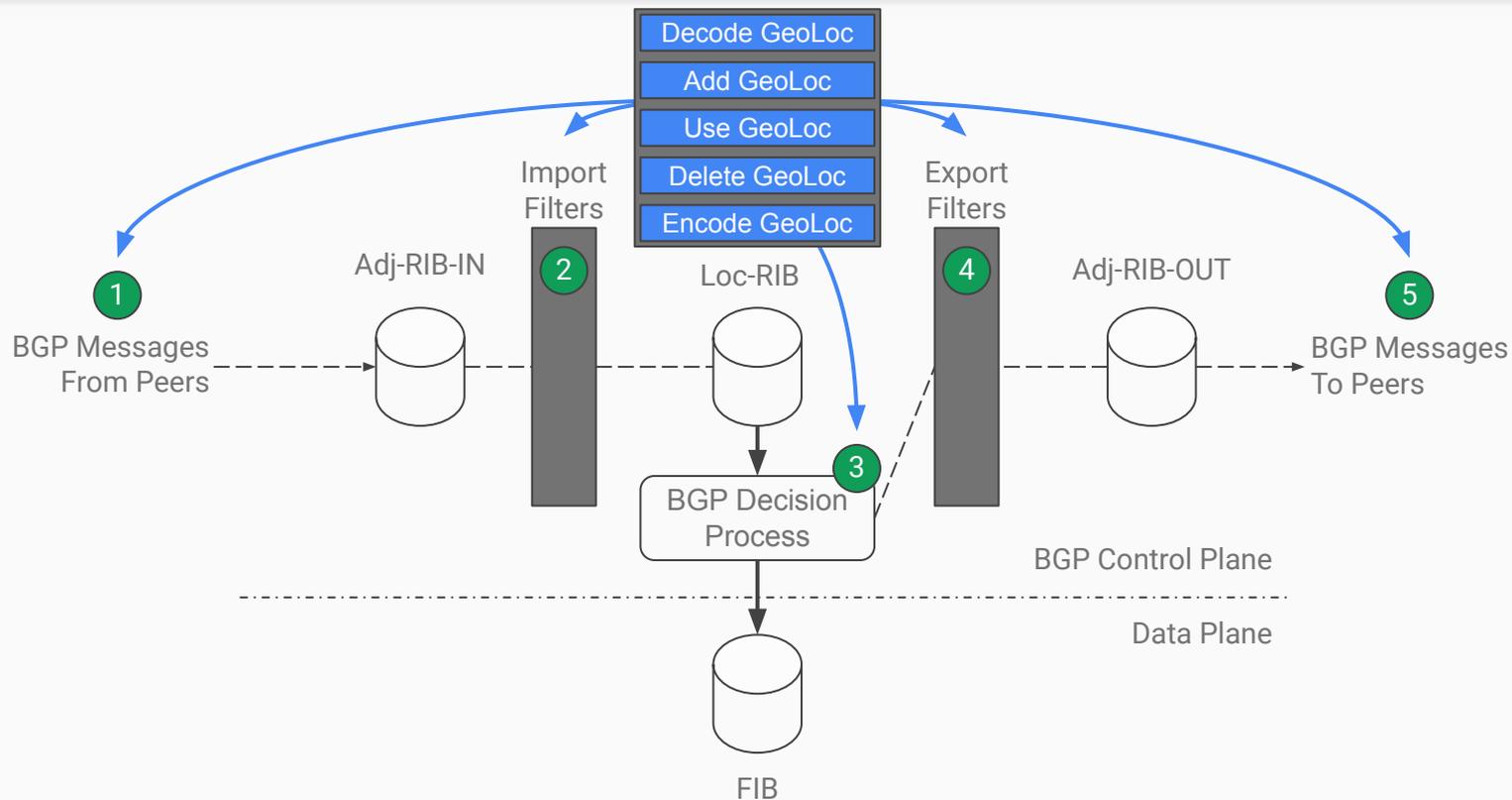
GeoLoc needs to alter the BGP Workflow



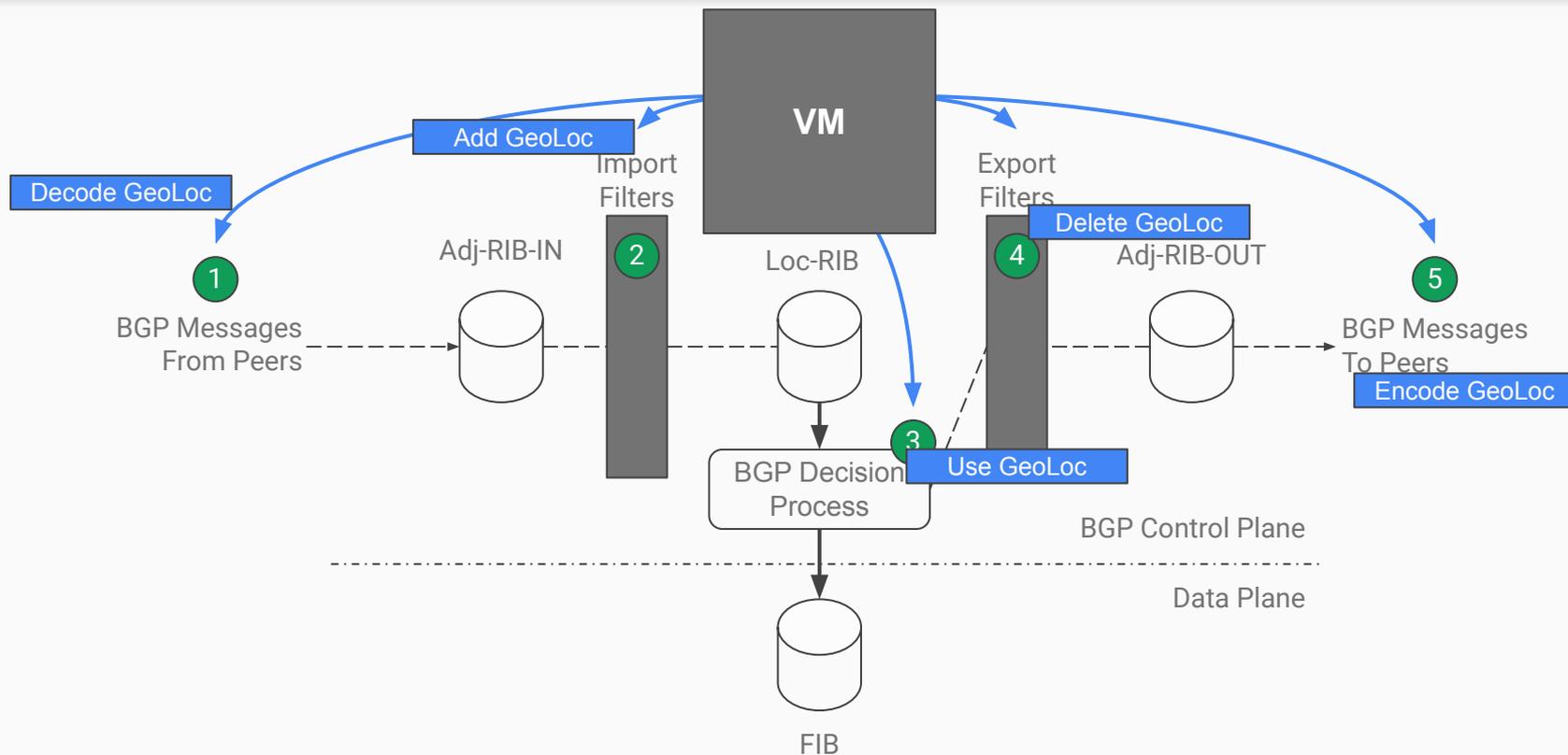
GeoLoc needs to alter the BGP Workflow



GeoLoc needs to alter the BGP Workflow

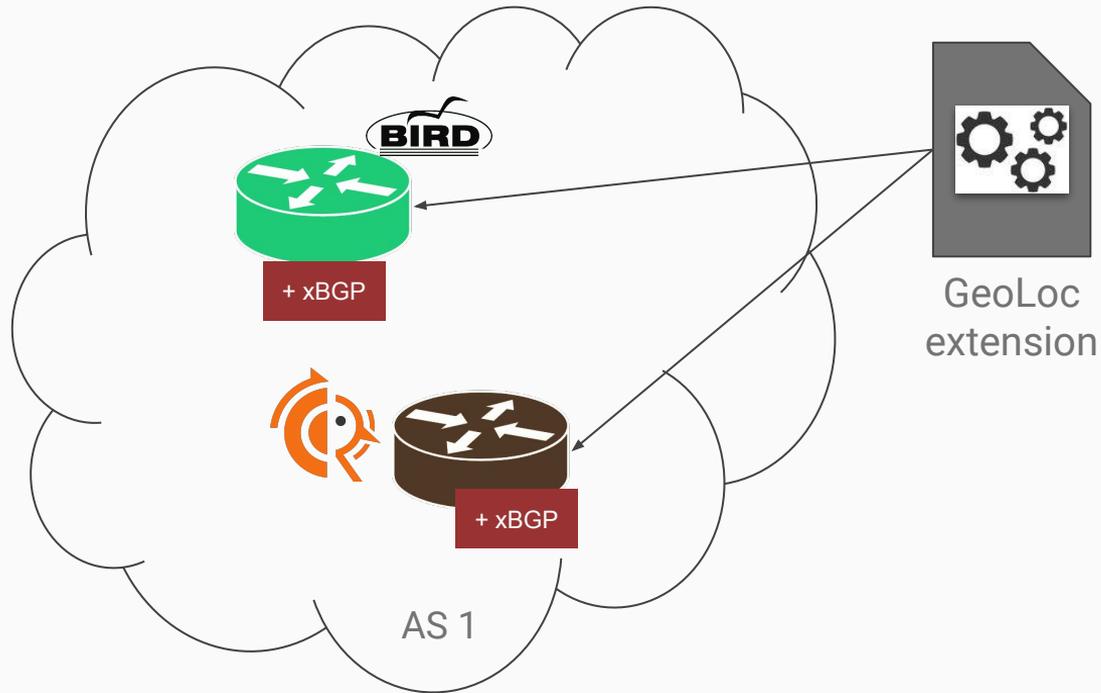


GeoLoc needs to alter the BGP Workflow



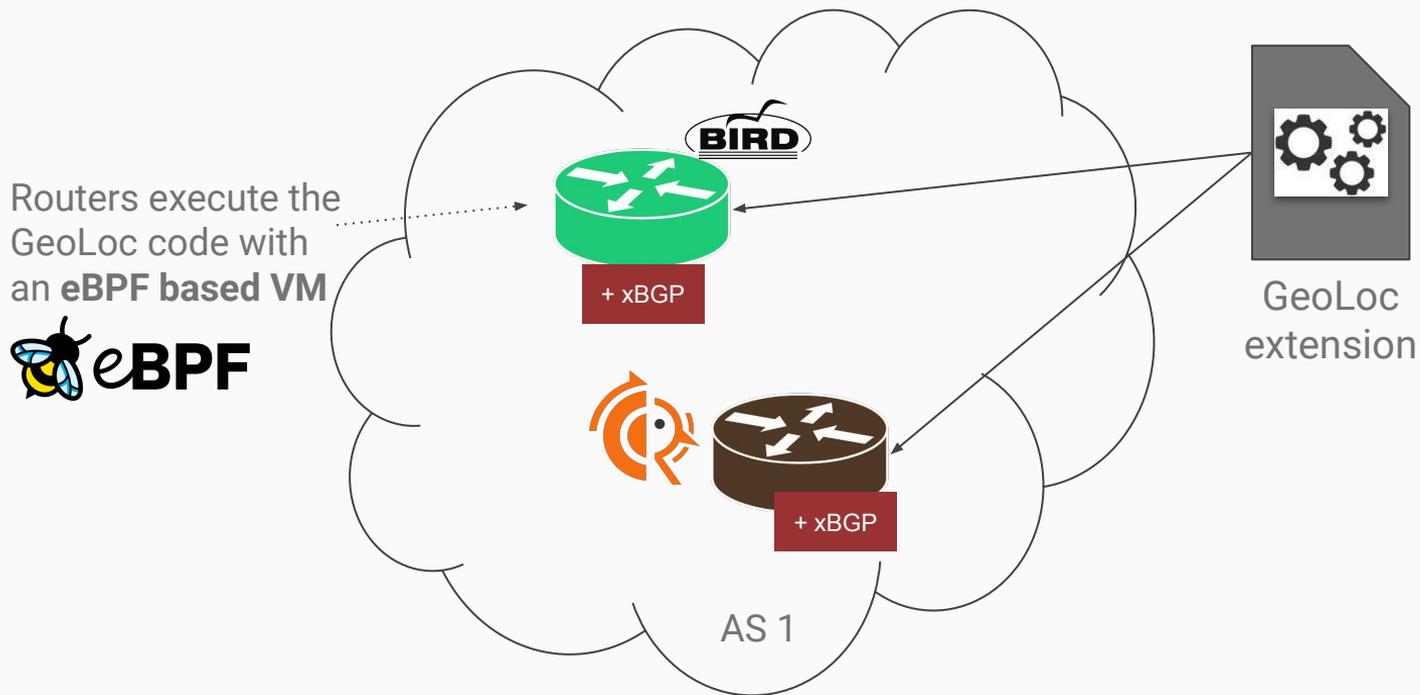
xBGP: a paradigm shift

Operators can now add extension codes to their routers



xBGP: a paradigm shift

Operators can now add extension codes to their routers



xBGP makes the link between Router and extensions

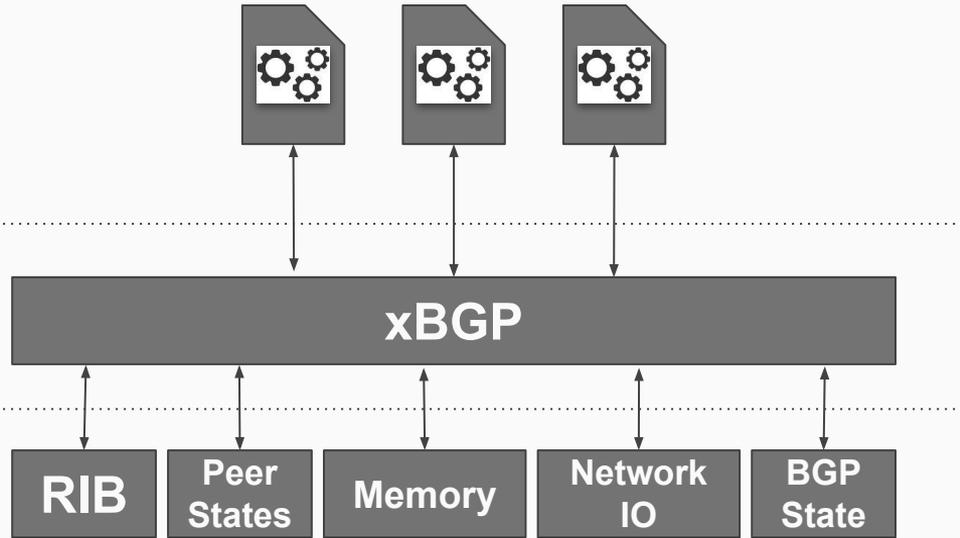
Provided by operators

xBGP Programs

Provided by our paper

Network OS Router

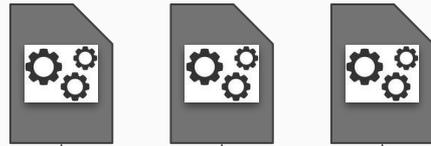
Provided by vendors



xBGP makes the link between Router and extensions

Provided by operators

xBGP Programs



Provided by our paper



Network OS Router



Provided by vendors



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
<code>libxbgp</code>	3004 + dependencies	
User Space eBPF VM	2776	

Other use cases

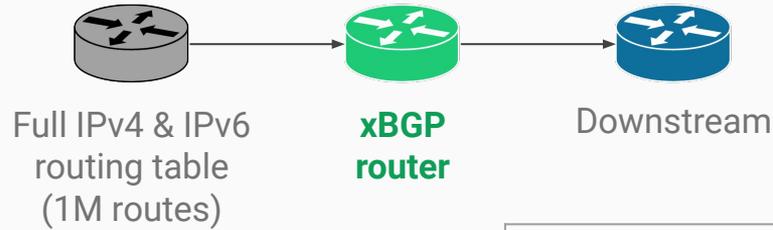
xBGP Extension	LoC
Geographical Location	388
Valley free routes	143
Filtering routes by IGP cost	36
Scanning for BGP zombies	1071
Influence remote BGP Decision Process	62
Monitoring the routes propagation time	806

⇒ Check the paper for those use cases

Agenda

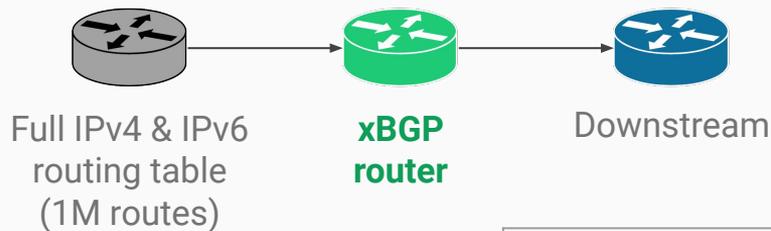
- Why bringing programmability to BGP ?
- Inside xBGP
- **Does using xBGP have an impact on router performances ?**
- Verifying xBGP extensions
- Conclusion

Using a Virtual Machine inside BGP



Use Case	Convergence Time	
	xFRR	xBIRD

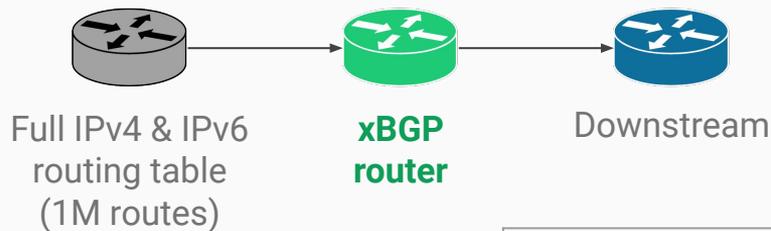
Using a Virtual Machine inside BGP



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

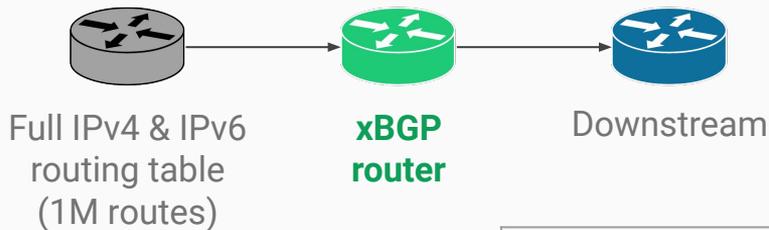


Additional overhead due to the xBGP internals

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%

Using a Virtual Machine inside BGP



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

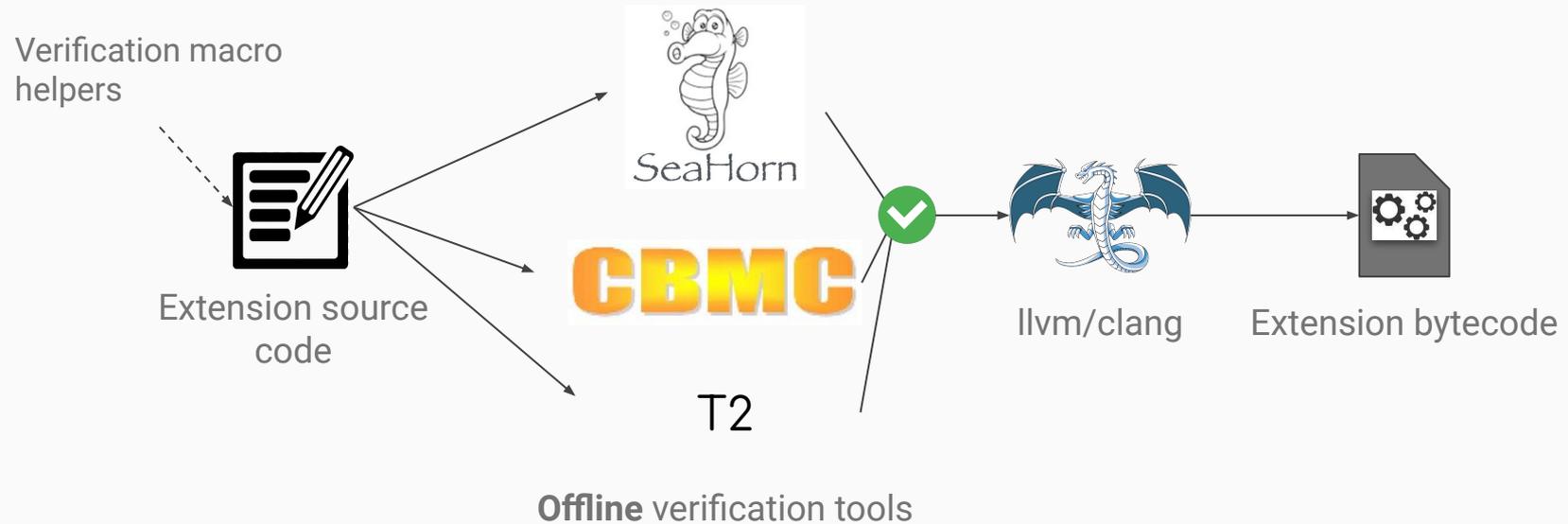
+ The “JIT” compiler is not efficient as native machine code

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**

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



The right tool to the right property

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

- **Seahorn**: BGP properties

Basic 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.

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|  Attr. Flags  |Attr. Type Code|
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|  Attr. Length (8 or 16 bits)  |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|  Latitude (64 bits)           |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|  Longitude (64 bits)         |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```

Verifying the BGP syntax of GeoLoc

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



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



`thomas.wirtgen@uclouvain.be`

[pluginized-protocols.org](https://www.pluginized-protocols.org)