



Monitoring a Telco

STEFFEN GEBERT

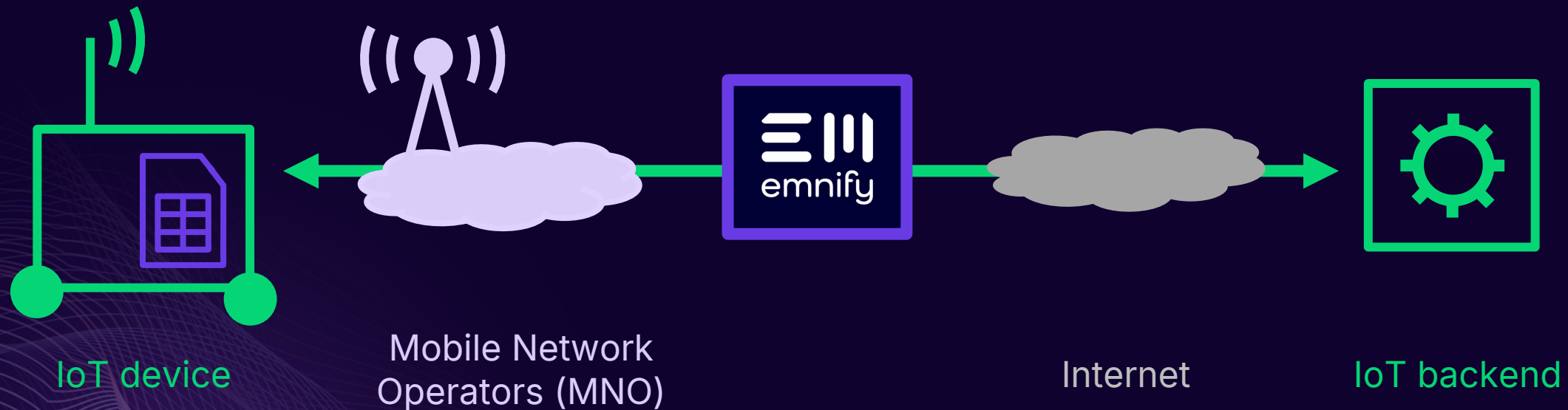
[ST-G.DE/SPEAKING](https://st-g.de/speaking)

SRECON EUROPE, MIDDLE EAST, AFRICA

DUBLIN, 14.10.2026



emnify IoT SuperNetwork



We're (usually) business critical for our customer



customer-owned



emnify-contracted



emnify-owned



nobody owns it

Things can break



We're (usually) business critical for our customer

What makes emnify different



IoT focus



We build and run our own mobile core



Globally distributed core network

The sad truth

We can *and must* do everything

What to expect from this talk



Cellular roaming



**Stream
processing**



**The story
around it**

What to expect from this talk

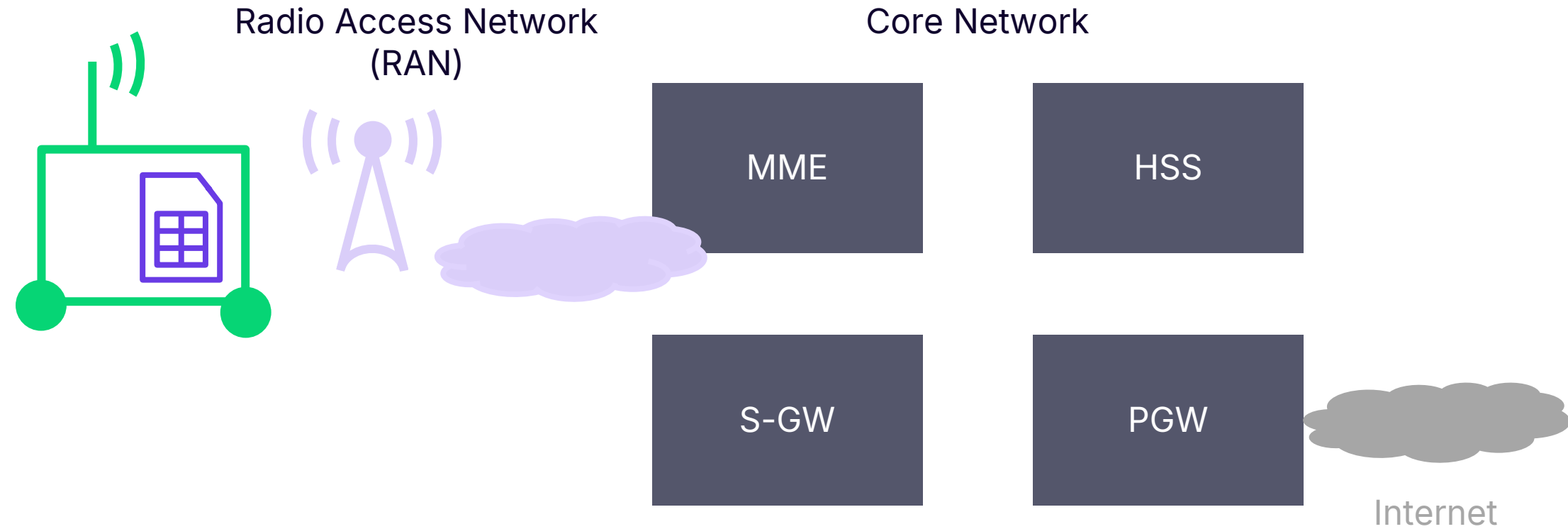


**The story
around it**

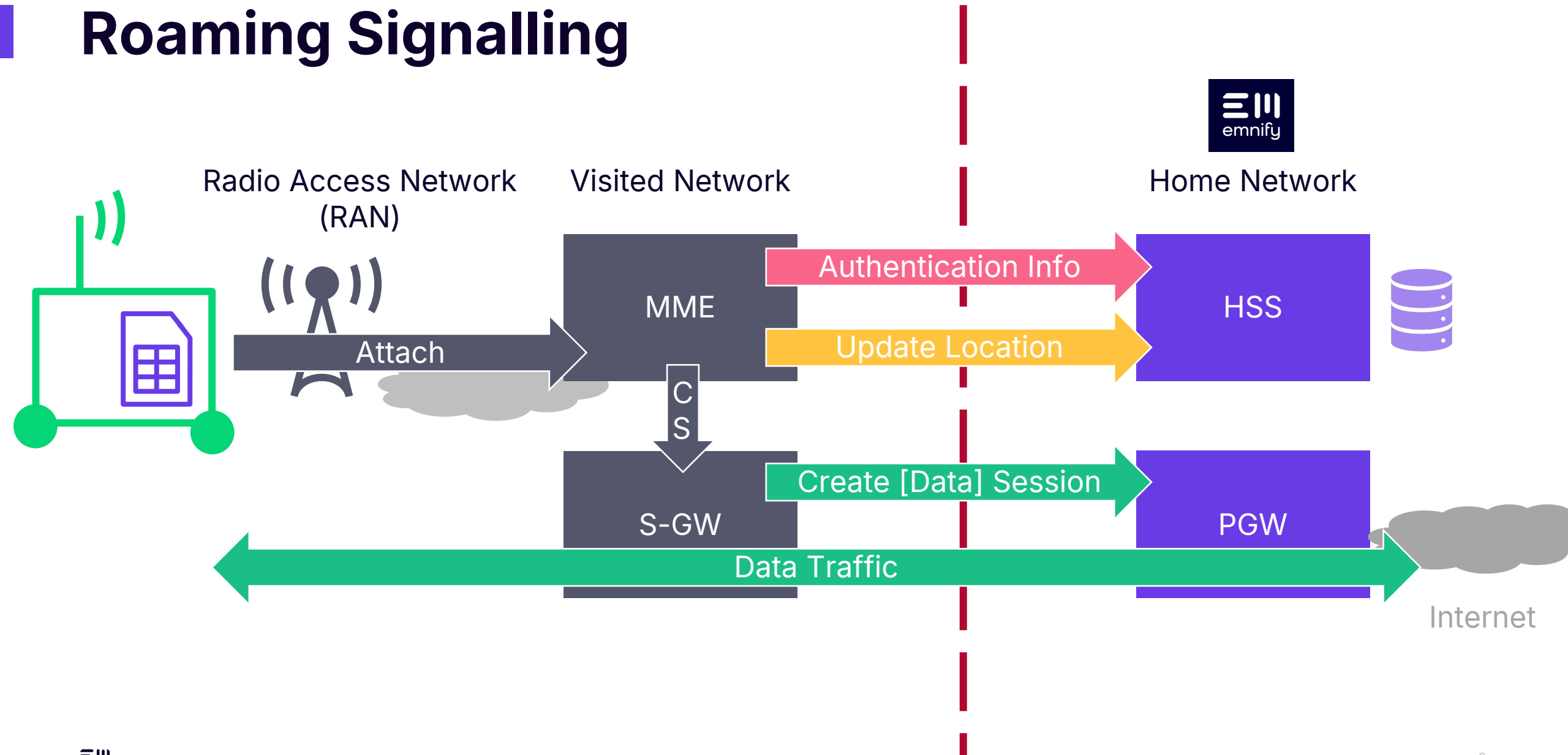


Cellular Networks & Roaming

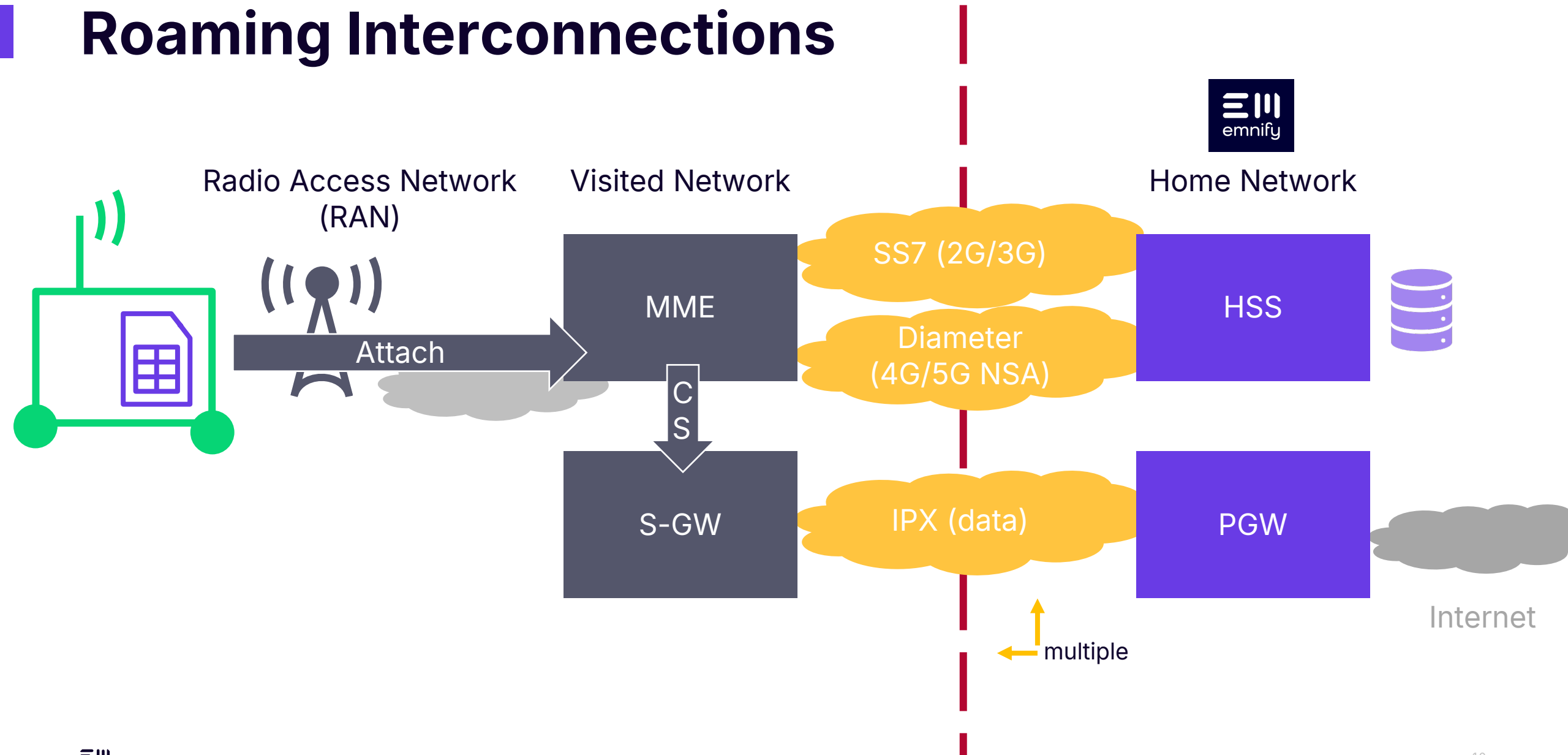
LTE Network Architecture



Roaming Signalling



Roaming Interconnections



Monitoring Approach

Application Performance Monitoring



Success & error rates
per request type



Per region, carrier
and pod



No customer or
operator info

Low
cardinality

Prometheus /
Mimir

Loki (Grafana
Cloud)

Signalling “Business” Events



Events emitted by applications

Kafka

Kafka Streams

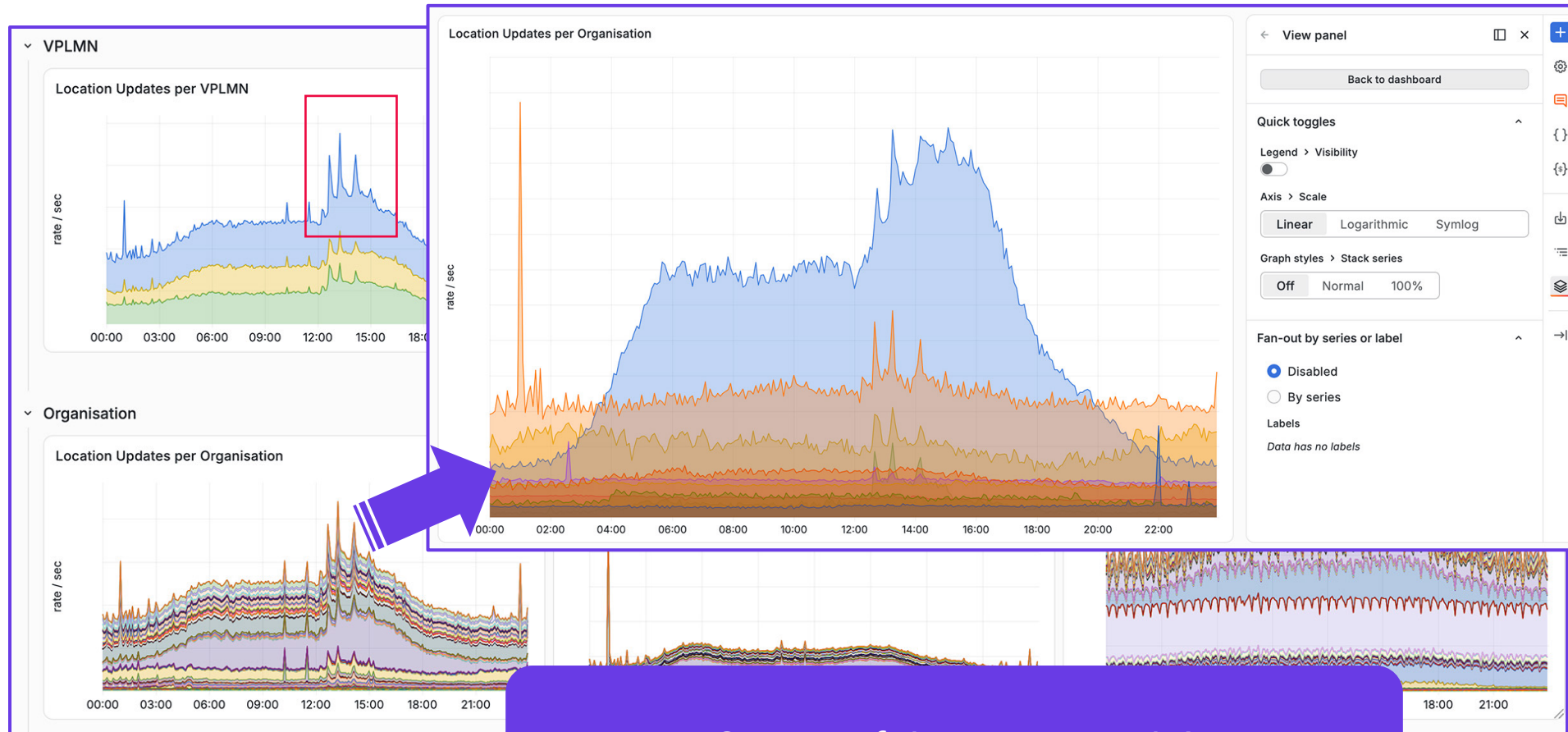
OpenSearch

My journey to Dublin

10:29:20 AM		Update location	 Italy
10:29:24 AM		Update GPRS location	 Italy
10:29:24 AM		Create PDP Context	 Italy
10:29:29 AM		Update GPRS location	 Italy
12:56:00 PM		Delete PDP Context	 Italy
2:51:28 PM		Update GPRS location	 Germany
2:51:30 PM		Update location	 Germany
2:51:31 PM		Create PDP Context	 Germany
3:22:31 PM		Delete PDP Context	 Germany
5:59:17 PM		Update GPRS location	 Ireland

Timestamp	Operation	SIM ID	Customer	Operator	Details	
14:00:01.123	Update Location	111	42	Vodakom	{...}	Update location  Ireland
14:00:01.124	Create Session	222	4321	O3	{...}	Update GPRS location  Ireland
						Create PDP Context  Ireland

Signalling "Business" Events



Successful requests only!

Kafka
Kafka Streams
Apache Iceberg on S3
Starrocks

The Monitoring Gap



Our Monitoring Gap

Understanding the impact is often harder than resolving the incident.



**0% errors, but
still broken**



**How many SIMs
create 1000 errors/s?**



**How long was a
device offline?**

What we need



**Full details, without
additional load**



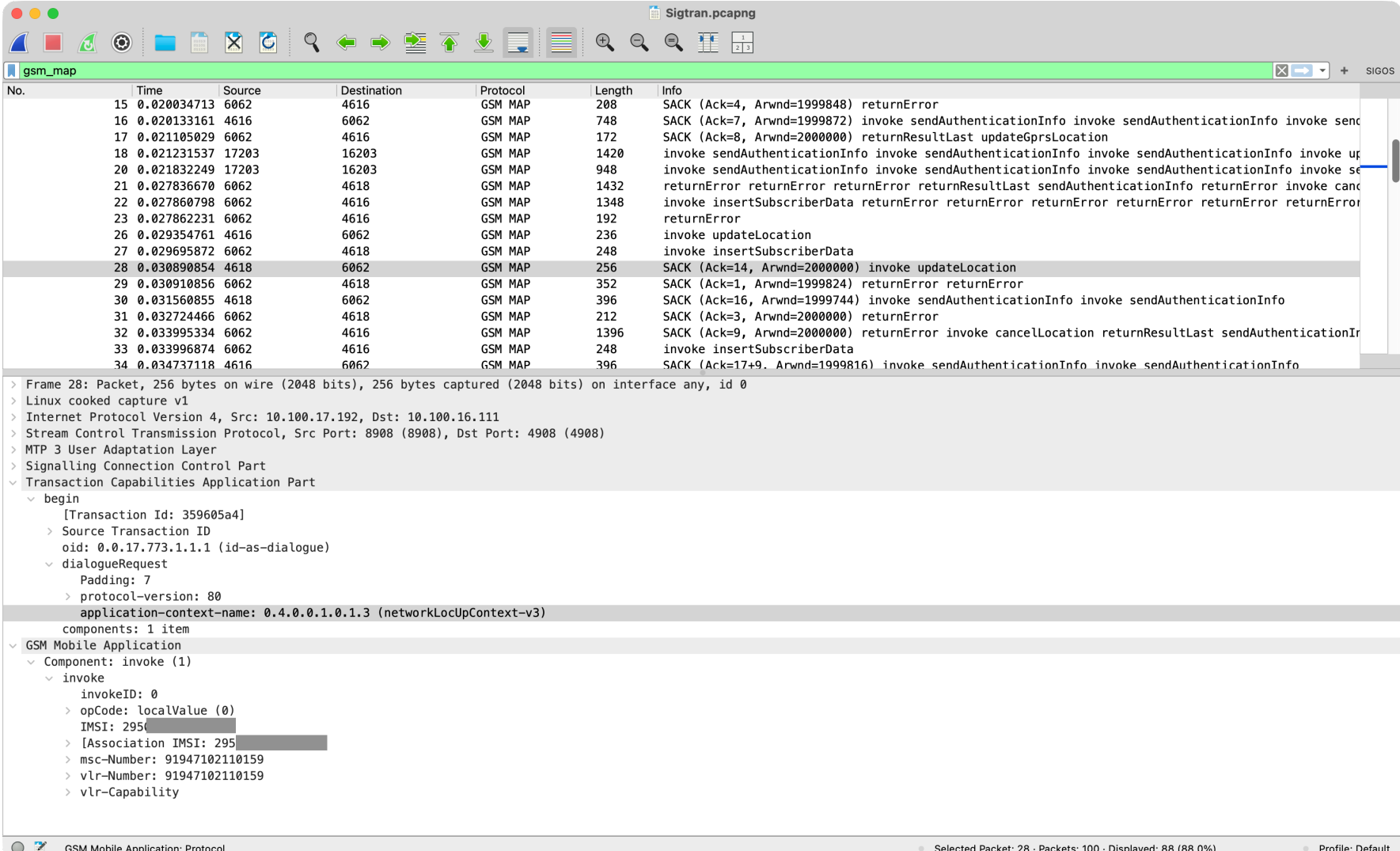
**Don't rely on the
application**



**Cover
"unseen errors"**

Go one level deeper.

Truth is on the wire



The image shows a Wireshark packet capture of a GSM MAP protocol session. The top pane displays a list of 34 packets. Packet 28 is selected, showing a SACK (Ack=14, Arwnd=2000000) and an invoke updateLocation. The bottom pane shows the detailed view of packet 28, including the Transaction Capabilities Application Part (TCAP) and the GSM Mobile Application (GSM MAP) components. The TCAP section shows a begin block with Transaction ID 359605a4, Source Transaction ID 0.0.17.773.1.1.1 (id-as-dialogue), and a dialogueRequest block with Padding: 7 and protocol-version: 80. The GSM MAP section shows a component: invoke (1) with an invoke block containing invokeID: 0, opCode: localValue (0), IMSI: 295, [Association IMSI: 295], msc-Number: 91947102110159, vlr-Number: 91947102110159, and vlr-Capability.

No.	Time	Source	Destination	Protocol	Length	Info
15	0.020034713	6062	4616	GSM MAP	208	SACK (Ack=4, Arwnd=1999848) returnError
16	0.020133161	4616	6062	GSM MAP	748	SACK (Ack=7, Arwnd=1999872) invoke sendAuthenticationInfo invoke sendAuthenticationInfo invoke sendAuthenticationInfo
17	0.021105029	6062	4616	GSM MAP	172	SACK (Ack=8, Arwnd=2000000) returnResultLast updateGprsLocation
18	0.021231537	17203	16203	GSM MAP	1420	invoke sendAuthenticationInfo invoke sendAuthenticationInfo invoke sendAuthenticationInfo invoke sendAuthenticationInfo
20	0.021832249	17203	16203	GSM MAP	948	invoke sendAuthenticationInfo invoke sendAuthenticationInfo invoke sendAuthenticationInfo invoke sendAuthenticationInfo
21	0.027836670	6062	4618	GSM MAP	1432	returnError returnError returnError returnResultLast sendAuthenticationInfo returnError invoke cancelLocation
22	0.027860798	6062	4616	GSM MAP	1348	invoke insertSubscriberData returnError returnError returnError returnError returnError returnError returnError
23	0.027862231	6062	4616	GSM MAP	192	returnError
26	0.029354761	4616	6062	GSM MAP	236	invoke updateLocation
27	0.029695872	6062	4618	GSM MAP	248	invoke insertSubscriberData
28	0.030890854	4618	6062	GSM MAP	256	SACK (Ack=14, Arwnd=2000000) invoke updateLocation
29	0.030910856	6062	4618	GSM MAP	352	SACK (Ack=1, Arwnd=1999824) returnError returnError
30	0.031560855	4618	6062	GSM MAP	396	SACK (Ack=16, Arwnd=1999744) invoke sendAuthenticationInfo invoke sendAuthenticationInfo
31	0.032724466	6062	4618	GSM MAP	212	SACK (Ack=3, Arwnd=2000000) returnError
32	0.033995334	6062	4616	GSM MAP	1396	SACK (Ack=9, Arwnd=2000000) returnError invoke cancelLocation returnResultLast sendAuthenticationInfo
33	0.033996874	6062	4616	GSM MAP	248	invoke insertSubscriberData
34	0.034737118	4616	6062	GSM MAP	396	SACK (Ack=17+9, Arwnd=1999816) invoke sendAuthenticationInfo invoke sendAuthenticationInfo

> Frame 28: Packet, 256 bytes on wire (2048 bits), 256 bytes captured (2048 bits) on interface any, id 0

- > Linux cooked capture v1
- > Internet Protocol Version 4, Src: 10.100.17.192, Dst: 10.100.16.111
- > Stream Control Transmission Protocol, Src Port: 8908 (8908), Dst Port: 4908 (4908)
- > MTP 3 User Adaptation Layer
- > Signalling Connection Control Part
- > Transaction Capabilities Application Part
 - > begin
 - [Transaction Id: 359605a4]
 - > Source Transaction ID
 - oid: 0.0.17.773.1.1.1 (id-as-dialogue)
 - > dialogueRequest
 - Padding: 7
 - > protocol-version: 80
 - application-context-name: 0.4.0.0.1.0.1.3 (networkLocUpContext-v3)
 - components: 1 item
 - > GSM Mobile Application
 - > Component: invoke (1)
 - > invoke
 - invokeID: 0
 - > opCode: localValue (0)
 - IMSI: 295
 - > [Association IMSI: 295]
 - > msc-Number: 91947102110159
 - > vlr-Number: 91947102110159
 - > vlr-Capability

Applying Stream Processing to Signalling Traffic



Let's develop our Wireshark replacement



Parse PCAP files
continuously



Process and
emit to data lake



Treat packets on the
wire as events



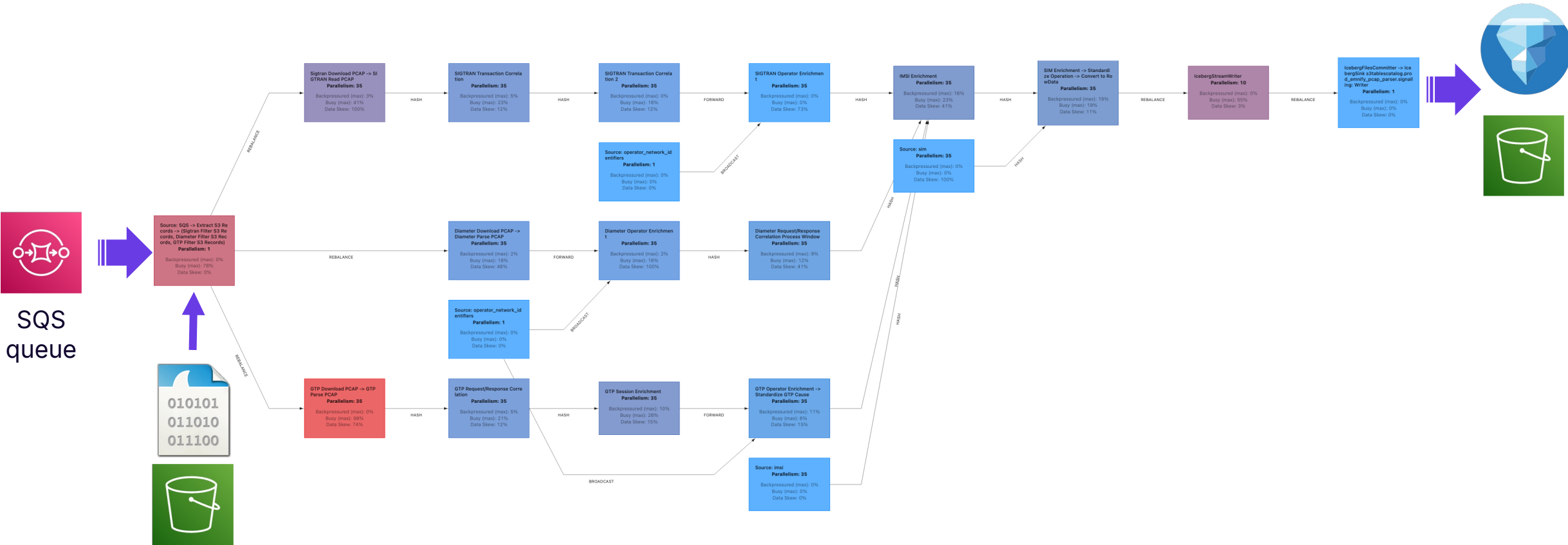
Flink

Meet **Apache Flink**

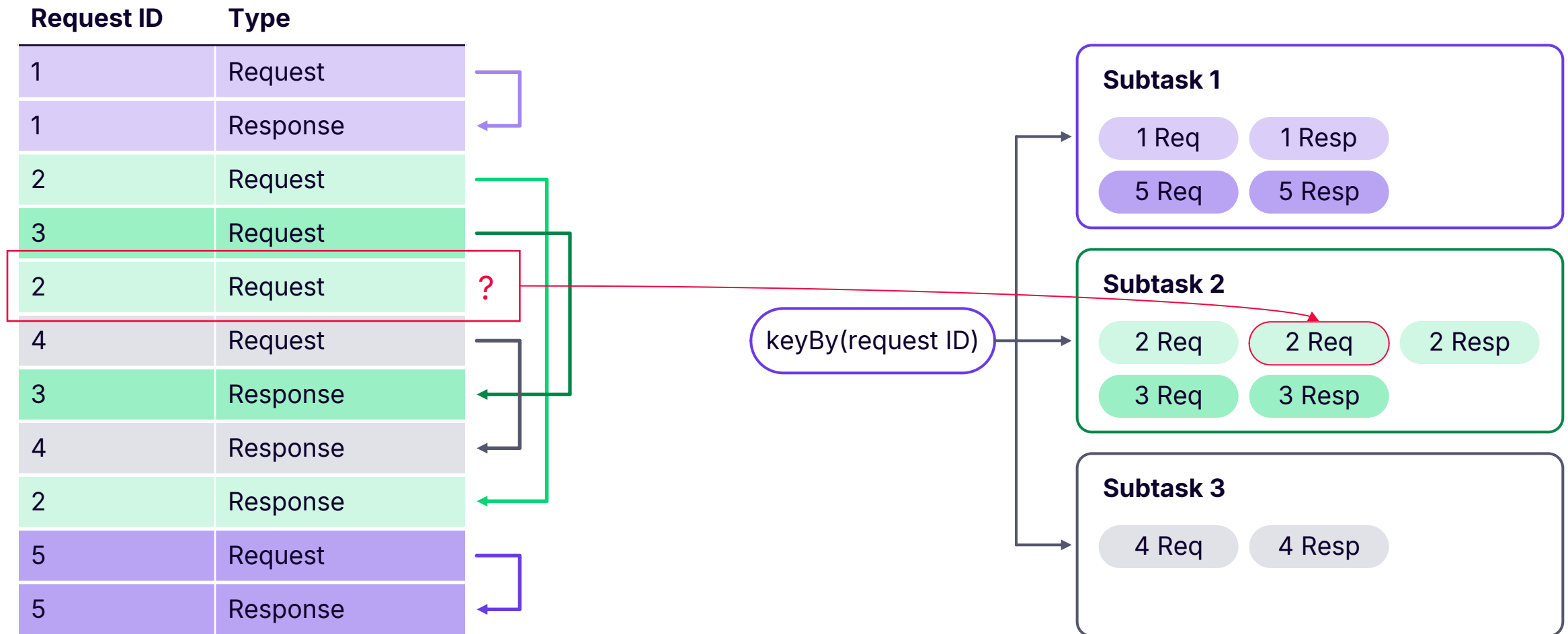
pkts for PCAP
parsing

jss7, jDiameter,
custom GTP parser

Flink's Data Flow Graph



Request / Response Correlation



Challenges



Retries



Overlapping
request IDs

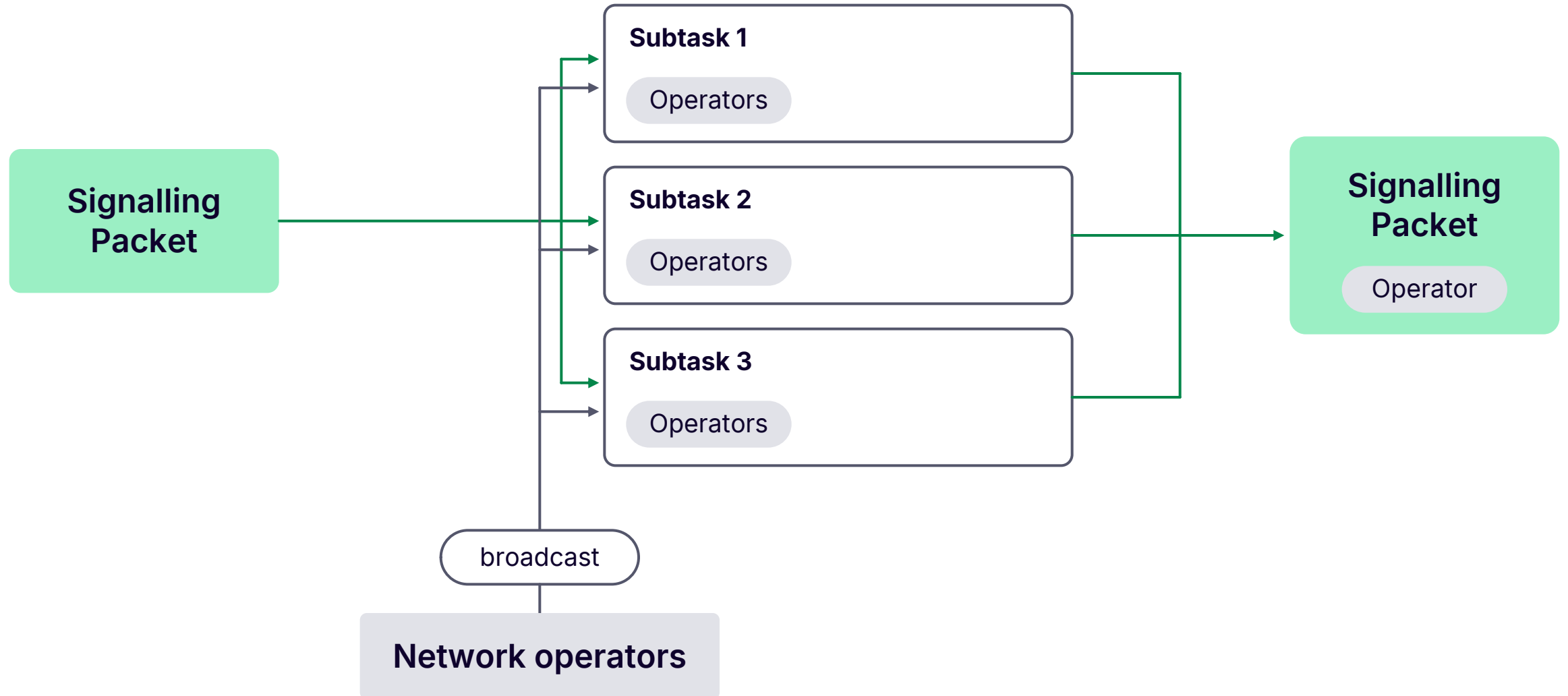


Lack of response

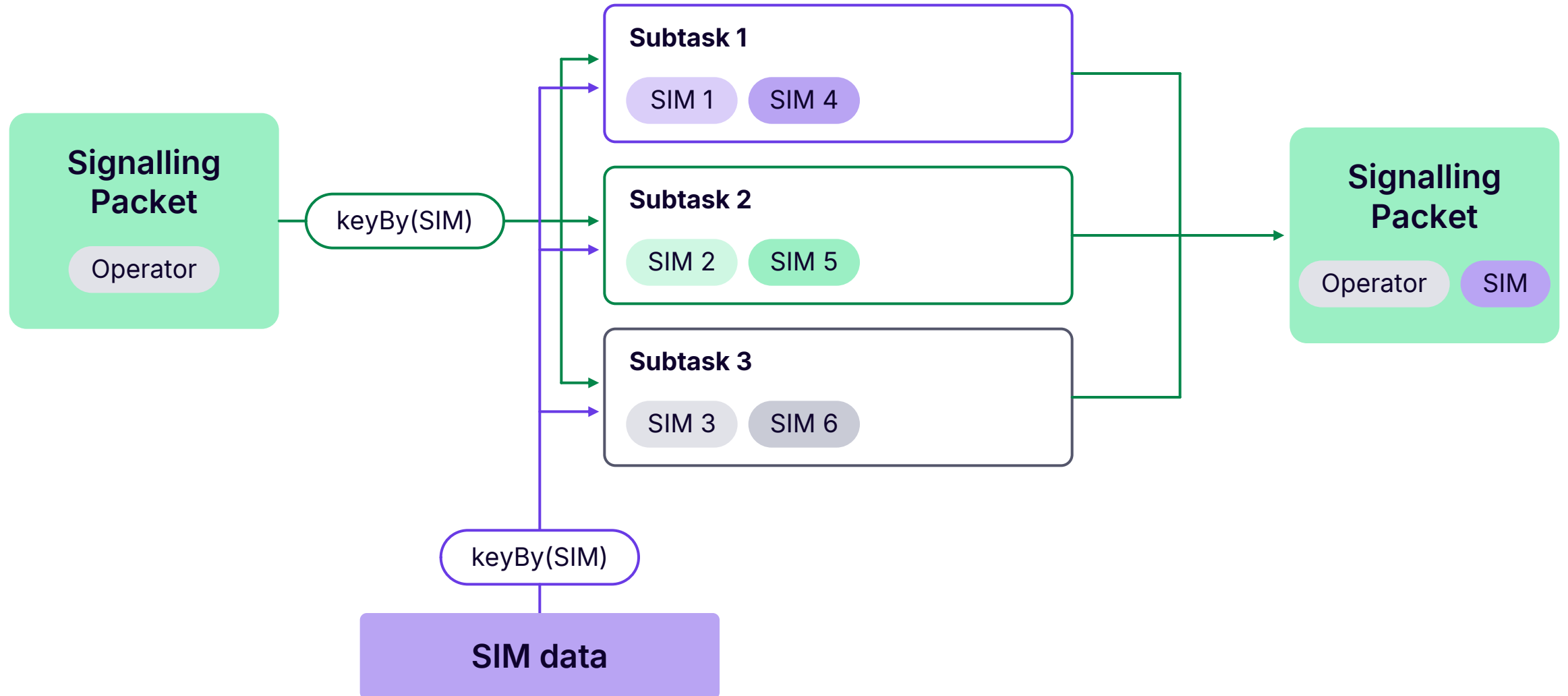
Flink's answer: window functions



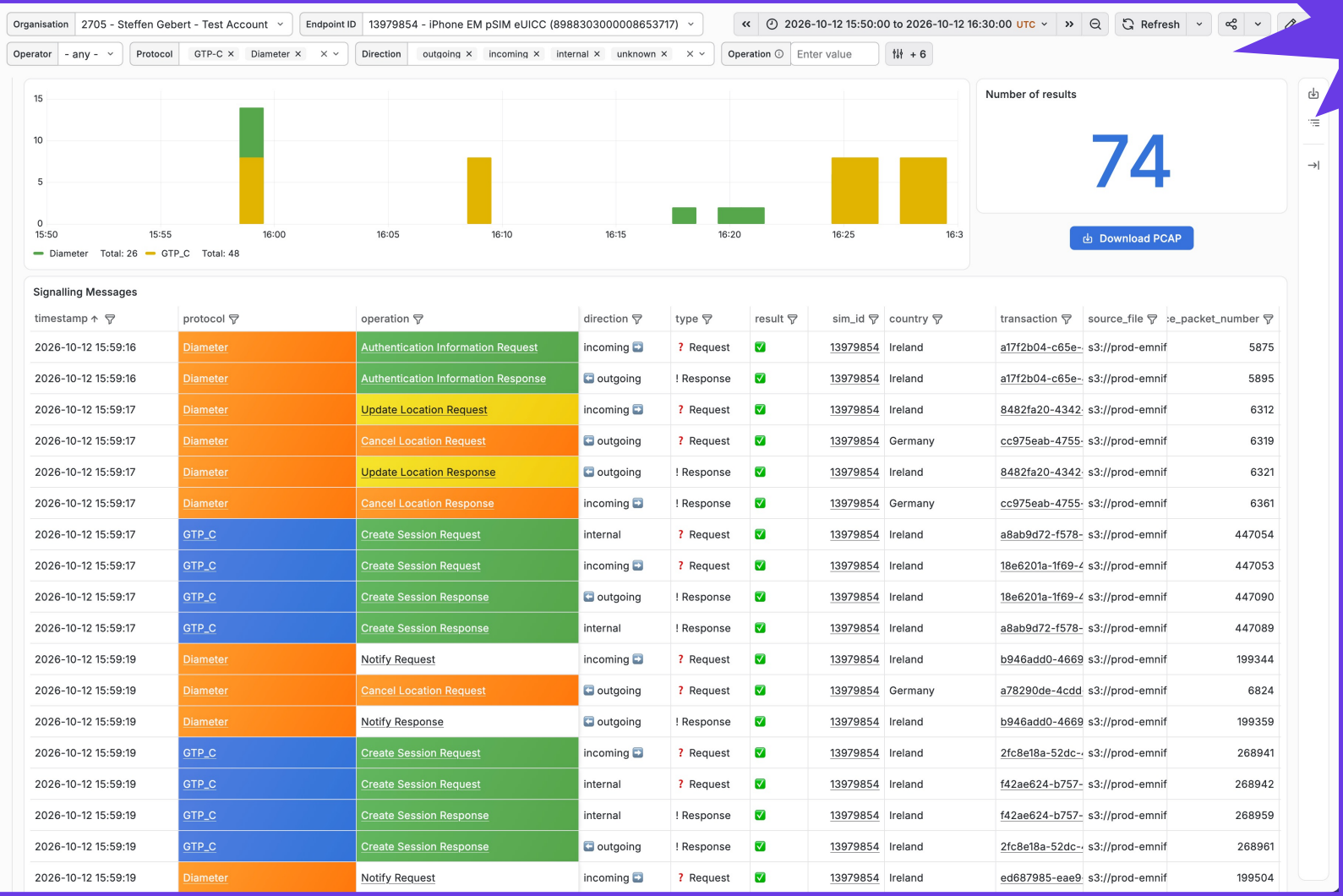
Enrichment: Network Operators



Enrichment: SIMs



Meet PCAP Parser

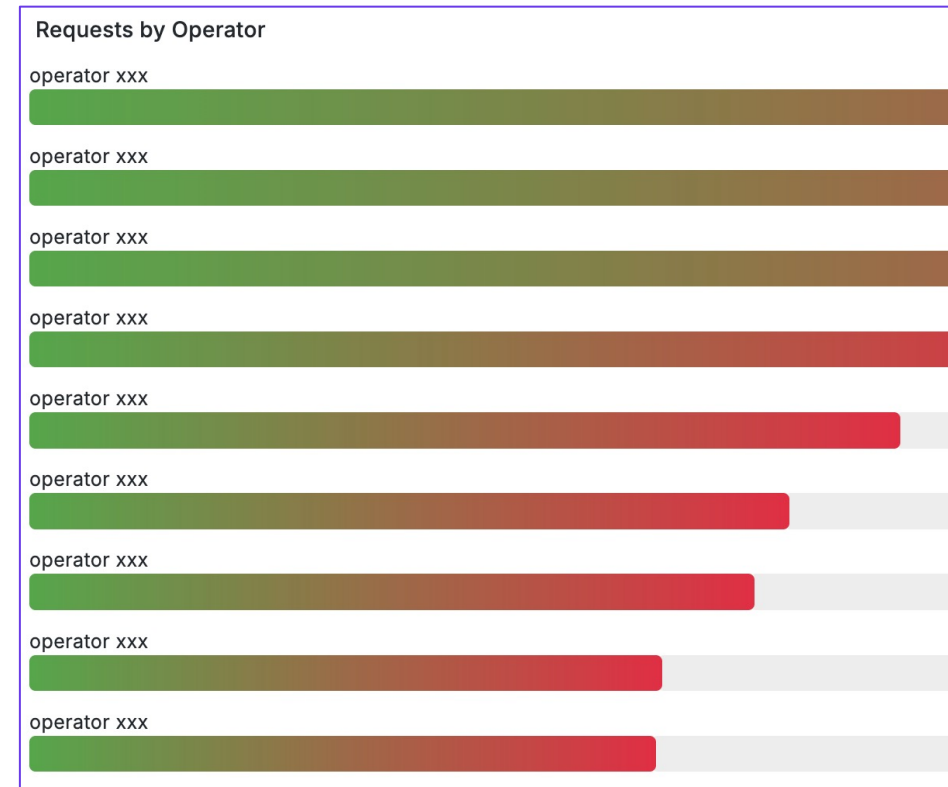


Grafana

S3 Tables
(Iceberg)

Starrocks

Analytics for Packets – Have Fun with SQL





The Non-technical Part

THE STORY AROUND IT

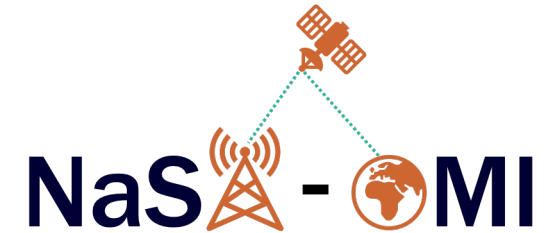


Research Projects

Running Research projects since 2019

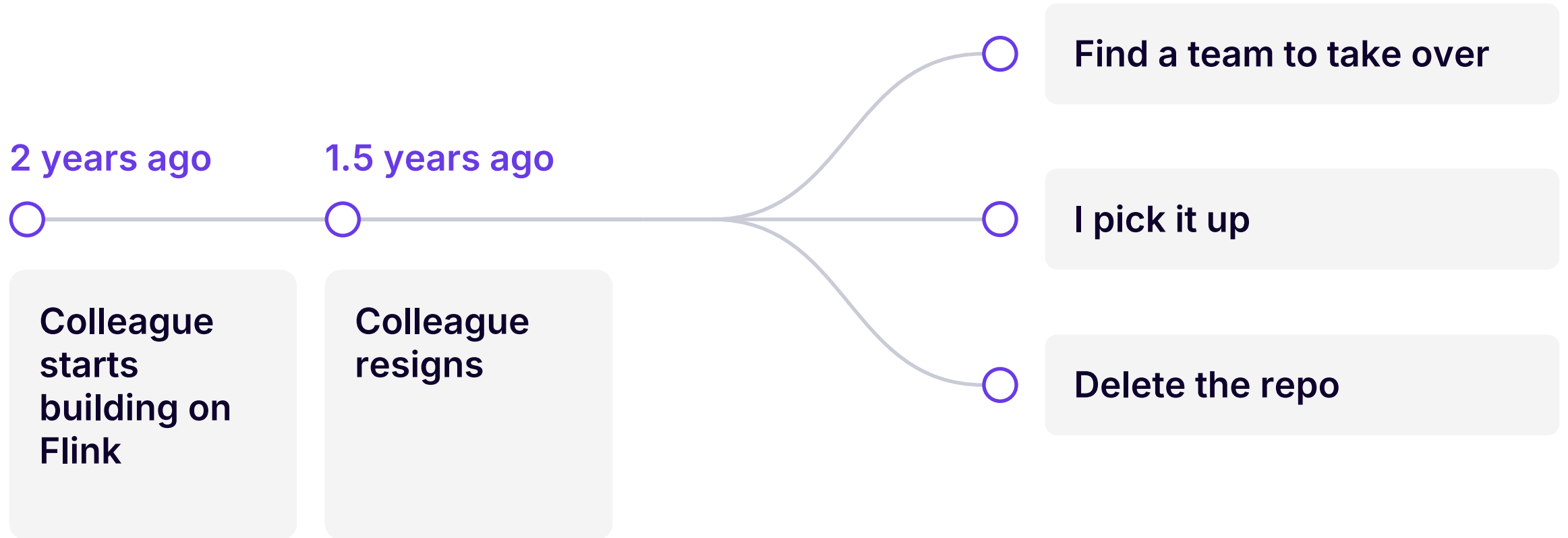
Universities are happy to hear from YOU!

Foundations for PCAP parser years ago

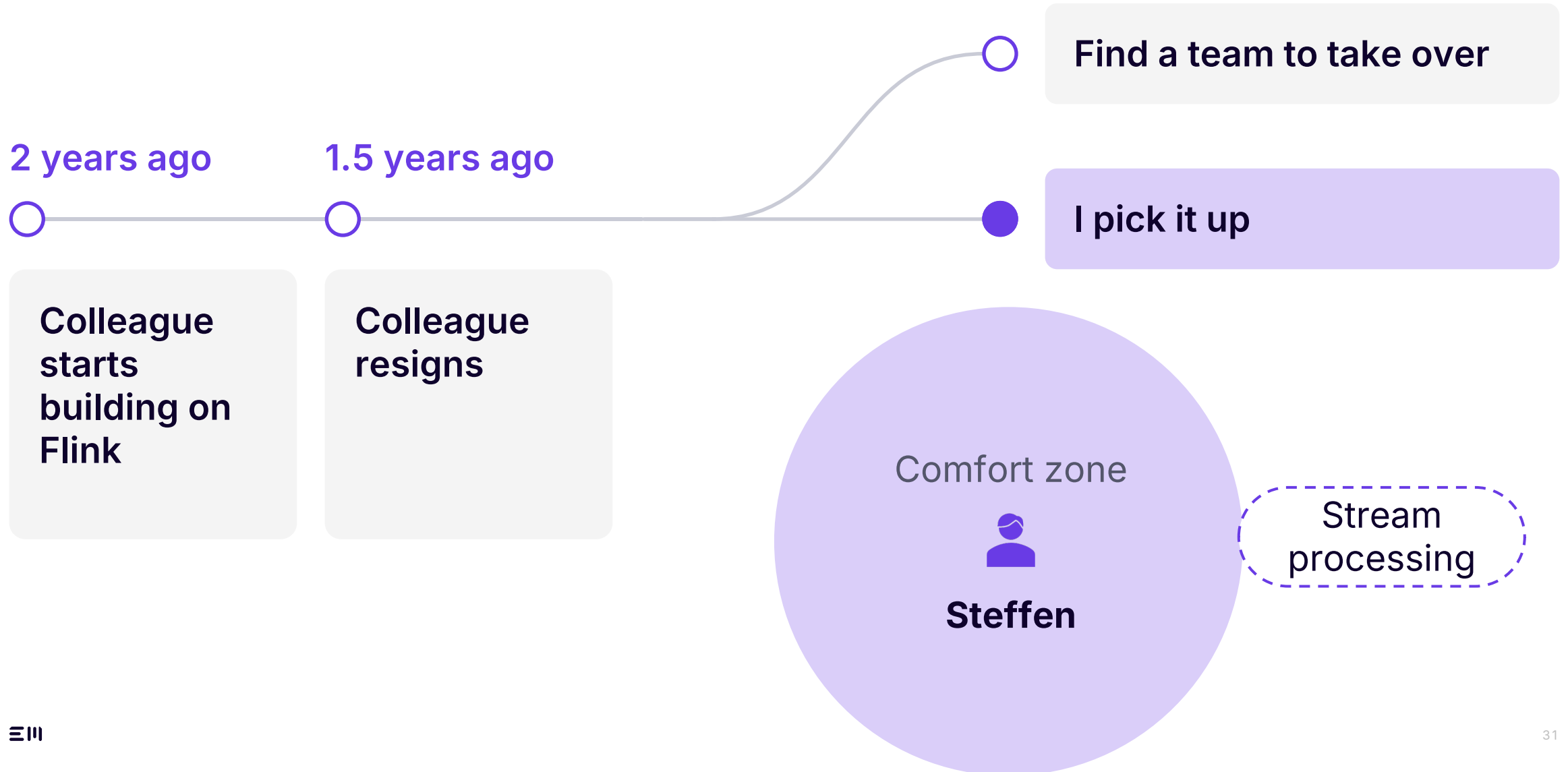


ORIGAMI project has received funding from the Smart Networks and Services Joint Undertaking (SNS JU) under the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101139270

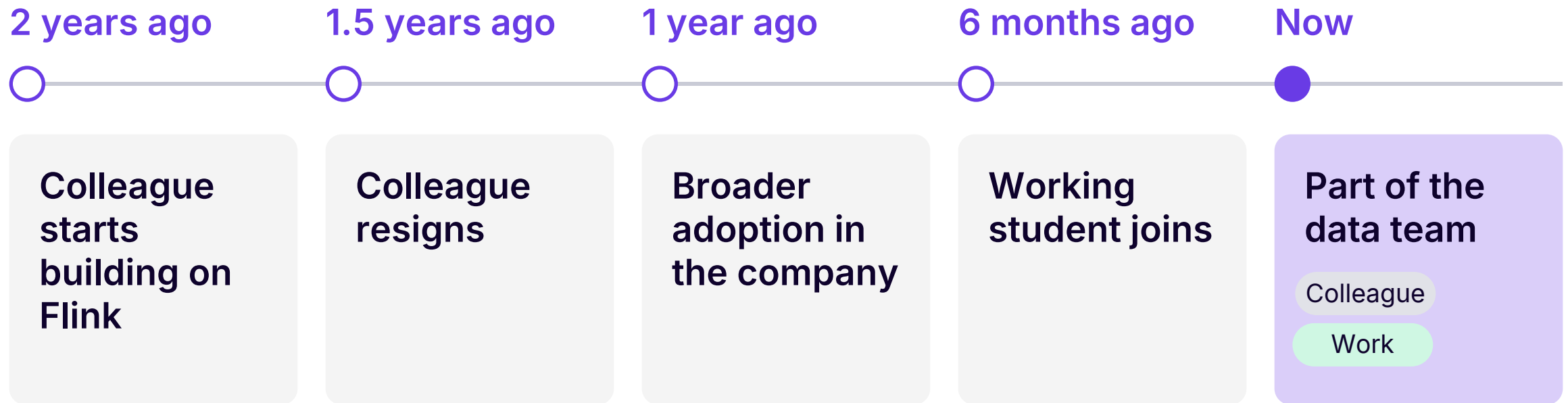
How I got there



How I got there



How we got there



Conclusion

Next Level for our Monitoring

- Required us to go down to raw data
- Packets for us – something else for you
- Applied methodologies that we already used elsewhere



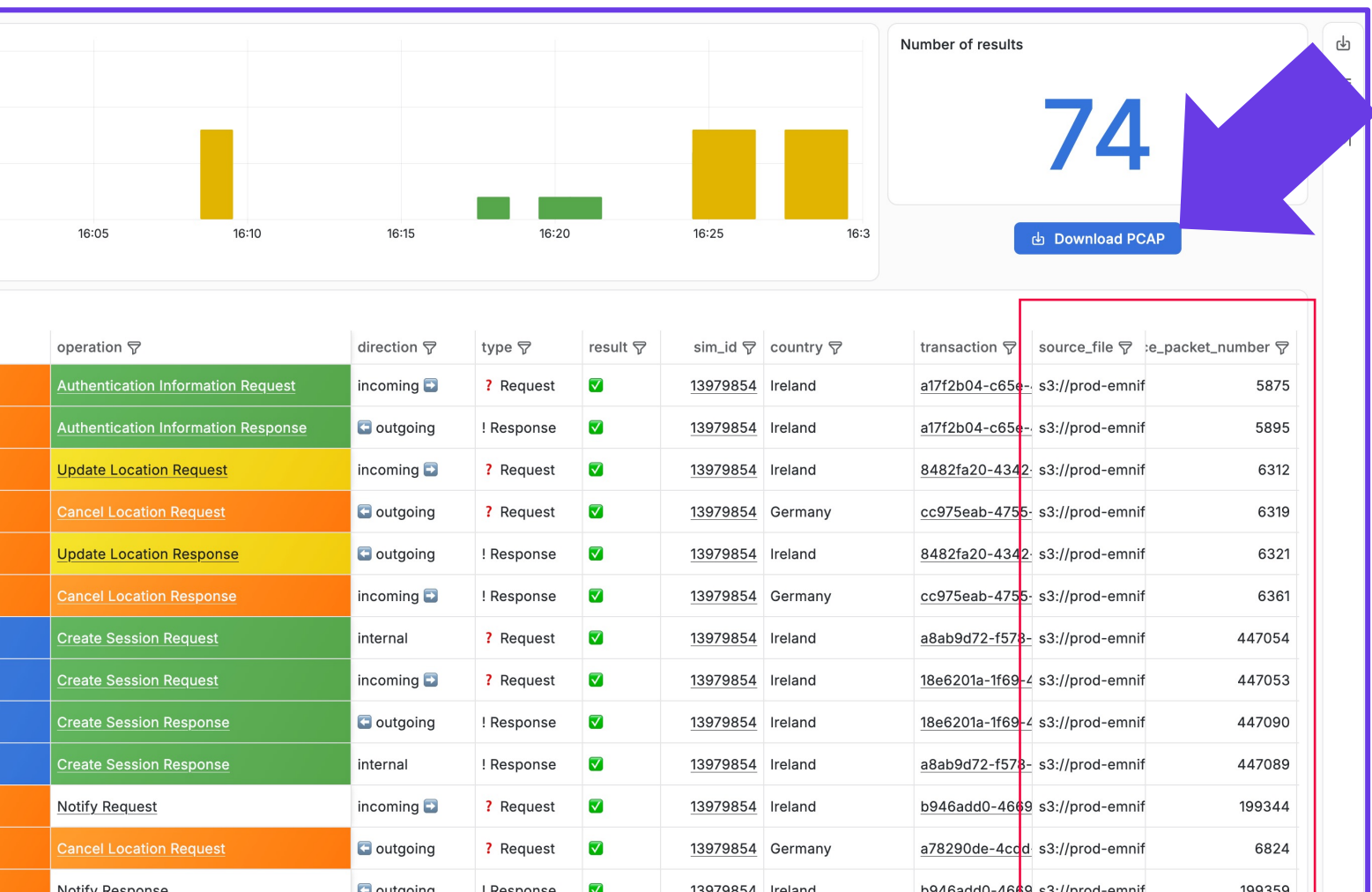
How we got there

- Required me to get out of my comfort zone
- Challenge to prioritize internal tooling covering multiple teams
- Research helps me to build foundations



Bonus Challenge

Users Asking for a Download Button



PCAP Extractor Grafana Plugin

