

- Introduction [5 min]
 - Sarah Edwards, GPO
- Bottom-Up
 - FOAM: New OpenFlow Aggregate Manager [10 min]
 - Josh Smift, GPO
 - Using SNAPP to Find and Visualize GENI Monitoring Data [20 min]
 - Camilo Viecco, Indiana University & John Meylor, Indiana University
 - Topology [15 min]
 - Chaos Golubitsky, GPO
- Top-Down
 - Monitoring/Mgmt Requirements & Discussion [1 hour]
 - Sarah Edwards, GPO

Monitoring & Management Requirements

Sarah Edwards, GPO

- Introduction
- Breakdown of Top Level Requirements
- Data Requirements
- Conclusions

- There is time built in to discuss as we go.

INTRODUCTION

- Would like to ensure:
 - We don't ignore any important pieces
 - Architecture decisions reflect needs of monitoring so that GENI Clearinghouse, I&M, etc serve our needs
 - Where possible, we build tools which can be adapted to new software when it becomes available
- Therefore we need to answer the following questions:
 - 1) Are there any holes in our understanding of our requirements?
 - 2) Do we have agreement on what we need in a way that can be communicated to other groups working on topics of interest to ops?
 - 3) Do we know how to build tools in an adaptable way?
- And also...
 - 4) What do we work on next?

We'll come back to these questions at the end

- 10.2-3 Visible operational status
 - The GENI system shall make sufficient data available that researchers and maintainers will be able to evaluate the availability and **operational** status of the system.
- 10.2-5 Federated event escalation
 - The GENI system shall provide operations and **management** support for event **management** and escalation, including security events, within GENI and with those organizations that interconnect with GENI.
- 10.2-6 Federated operations data exchange
 - The GENI system shall support operational and **management** data exchange according to [TBS] GENI O&M Policy between GENI and operators/owners of federated components, aggregates, and networks.

- Meta-operations (a.k.a. GMOC)
- Aggregates
 - Examples: ProtoGENI, PlanetLab, Orca
- Campuses
 - Which host & run aggregates
 - Which only host aggregates
- Backbone & Regional Networks
 - Networks which are GENI participants: I2, NLR
 - Networks which carry GENI traffic: some regionals (eg NoX)
- Experimenters

- Each GENI rack is a **SINGLE** aggregate
 - Therefore requirements are the same as for aggregates
- Aggregates can outsource (some of) their responsibilities to the GENI Clearinghouse

- Monitoring
 - Act of collecting data and measuring what is happening
- Management
 - Act of fixing problems and responding to requests
- What does monitoring & management involve?
 - Observe unexpected events
 - THEN fix what's wrong
 - Observe expected events
 - THEN develop policy for fixing what's wrong
 - THEN fix what's wrong (by responding to monitoring)
 - Plan for the future
 - Monitor long-term trends in resource usage
 - THEN provision resources to meet forecasted needs

What makes GENI different?

- Federated entities managed by different institutions with different policies
- People and information needed to troubleshoot and resolve problems are spread across several physical locations
- Users (end user and experimenters), managers and hosts of a given piece of equipment may all be different.
- Interactions between groups are governed by GENI federation agreements (e.g. aggregate provider agreement) and mutual understanding.

- We are not covering:
 - Monitoring and management which fits entirely within the purview of aggregates, campuses, etc
- For example, we will do (but not discuss here)
 - Keeping logs
 - Obeying local laws and policy
 - Answering the phone when someone has a concern
 - ... and tie your shoes and everything else.
- These things do **NOT** make GENI different

- Information must be shareable
- Information must be collected
- Information must be available when needed
- Cross-GENI operational statistics collected and synthesized to indicate GENI as a whole is working
- Preserve privacy of users (opt-in, experimenters, other users of resources)

- For both debugging and security problems:
 - Must be possible to escalate events
 - Meta-operations and aggregate operators must work together to resolve problems in a timely manner
- Must be possible to do an emergency stop in case of a problem
- Orgs must manage GENI resources consistent with local policy and best practices
 - e.g security procedures, logging, backups, etc
- Develop policies for monitoring
- All parties should implement agreed upon policies
- Security of GENI as a whole and its pieces

BREAKDOWN OF TOP LEVEL REQUIREMENTS

- *GENI monitoring is more than the sum of the monitoring at GENI's parts. In order to know if GENI is working properly, additional monitoring is required beyond that done by each of its constituent pieces.*
- Collect and synthesize additional operational statistics which indicate whether GENI is working
 - e.g. meso-scale ping tests, topology
 - Collect cross-GENI stats
 - Make cross-GENI stats available when needed

- Preserve privacy of users (opt-in, experimenters, other users of resources)
 - **→ TBD – This is an area needing major discussion**

- For both debugging and security problems:
 - Meta-operations and aggregate operators must work together to resolve problems
 - Aggregates must advertise resources accurately
 - (threshold) statically → Fill out aggregate page
 - (objective) dynamically → Advertise resources via AM API
 - Aggregates notify meta-operations when resources are unavailable → via e-mail (doing SOME of the time)
 - Aggregates cooperate with meta-operations on the resolution of security events
 - Aggregates cooperate with LLR on the resolution of security events
 - Must be possible to escalate events

- Must be possible to do an emergency stop in case of a problem
 - Must maintain POC information at meta-operations
 - Aggregate → send contact info to GMOC
 - Campus → send contact info to GMOC
 - Experimenter → slice e-mail
 - Other infrastructure → contacted by relevant campus
 - Aggregates & Meta-operations must each have policies and procedures in place to support an emergency stop
 - → Has been dry run

- Orgs must manage GENI resources consistent with local policy and best practices (e.g security procedures, logging, backups, etc)
 - In general, follow local policy and procedures
 - Follow best practices which if not followed would affect other members of the GENI community
- Develop policies for monitoring
- All parties should implement agreed upon policies
 - Follow Aggregate Provider Agreement
 - Follow LLR
 - Follow other GENI policies as they come into effect

- Security of GENI as a whole and its pieces
 - Two things we want to prevent:
 - Compromise of GENI resources
 - Use of GENI resources to compromise other entities
 - Two things we can do about this:
 - Follow best practices to hinder compromise
 - Detect and respond to compromise
 - Allow interesting research for which experimenters and operations have to coordinate for security and management reasons
 - Security experimentation BOF tonight and session at GEC13!!!
 - → **TBD – This is an area needing major discussion**

Info must be shareable/collected/available

- **Information must be shareable**
 - Consistent definitions of data
 - Consistent data exchange format
 - Consistent data collection mechanisms
 - Data sharing mechanisms
 - The following benefit from shared common processes:
 - Accessing data, finding data, visualizing data
- **Information must be collected**
 - Verify continued successful data collection
 - Debug collection and reliability outages
- **Information must be available when needed**
 - Privacy of data must be maintained

DATA REQUIREMENTS

- Consistent definition of data
 - Relational data
 - Resources (incl. connectivity)
 - List of aggregates
 - List of slices
 - List of users
 - Aggregate contact information
 - Timeseries data
 - Examples: Host and network statistics
 - Events
 - Examples: SNMP Traps

- Data collection methods
 - Relational data → store in relational DB
 - Resources → Rspecs available via AM API
 - List of aggregates → ctrl framework clearinghouse & GENI wiki
 - List of slices → control framework slice authority
 - List of users → TBD
 - Aggregate contact information → aggregate page and GMOC
 - Timeseries data → store in RRD
 - → collect via SNMP (ie host and network stats)
 - → by asking the aggregate (ie custom OpenFlow API)
 - Events → store in relational DB (?)
 - → TBD

- Sharing Data
 - → publish to central DB at GMOC
 - → publish locally via webpage or local API
 - → TBD: publish via a distributed mechanism
- Accessing, Finding and Visualizing Data
 - → GMOC Portals
 - → GMOC SNAPP Interface (with search)
 - → GMOC data available to interested consumers via API
 - → TBD: More to do here

- Troubleshooting info from aggregates, campuses, meta-operations
- Accountability report: How to prove if this is not my fault?

CONCLUSIONS

- Would like to ensure:
 - We don't ignore any important pieces
 - Architecture decisions reflect needs of monitoring so that GENI Clearinghouse, I&M, etc serve our needs
 - Where possible, we build tools which can be adapted to new software when it becomes available
- Therefore we need to answer the following questions:
 - 1) Are there any holes in our understanding of our requirements?
 - 2) Do we have agreement on what we need in a way that can be communicated to other groups working on topics of interest to ops?
 - 3) Do we know how to build tools in an adaptable way?
- And also...
 - 4) What do we work on next?

1) Are there any holes?

- From above:
 - Security
 - Privacy
 - Topology
 - Can stitching or other SW efforts help?
 - Accessing, finding and visualizing data
- From yesterday:
 - Topology
 - Event notification system
 - How does a third party help troubleshoot a slice?
 - Slice traceroute?
 - Universal names (eg for circuits)
 - need to coordinate with the software group

2) Agreement on requirements?

3) Building adaptable tools?

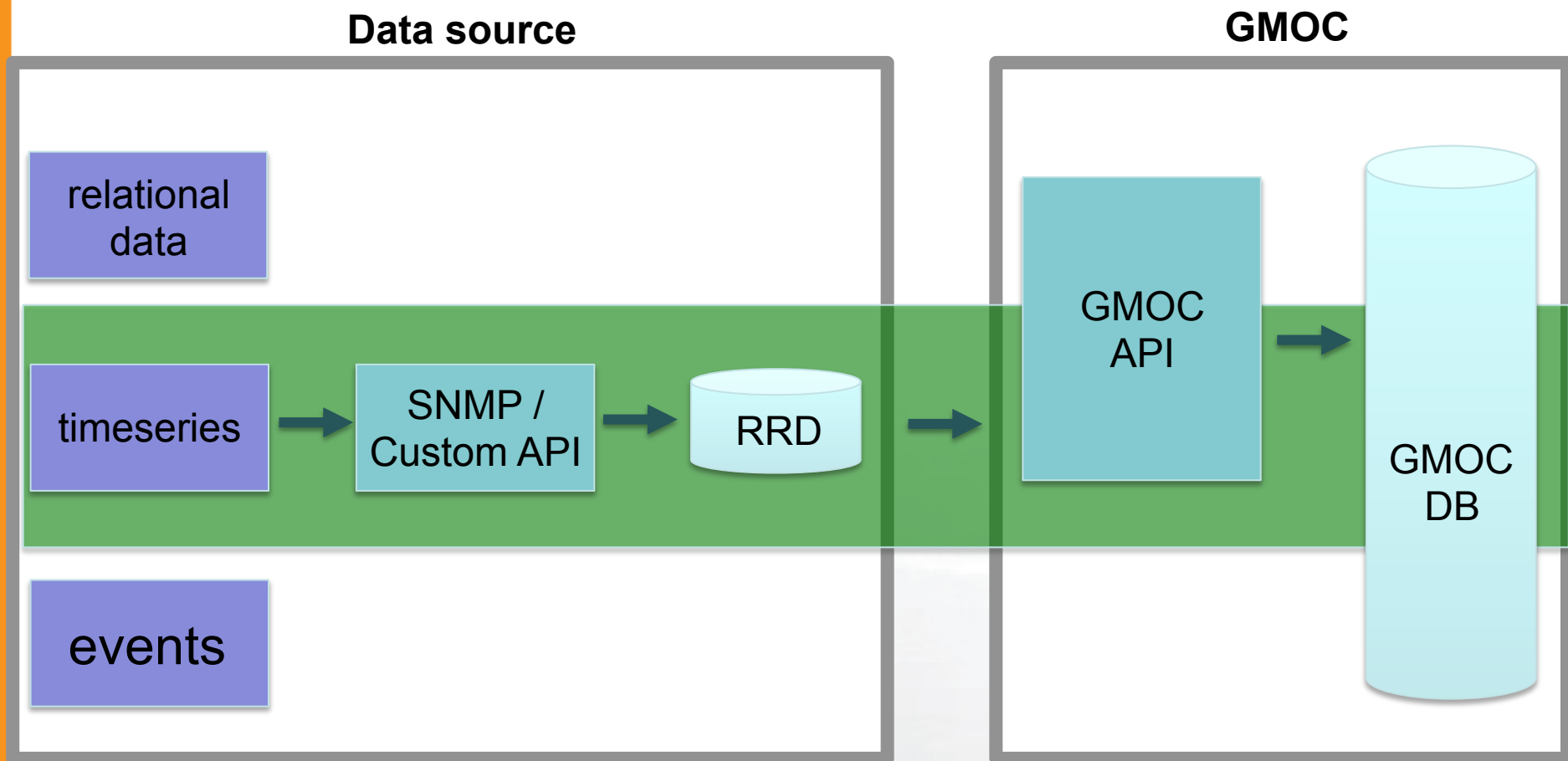
relational
data

timeseries

events

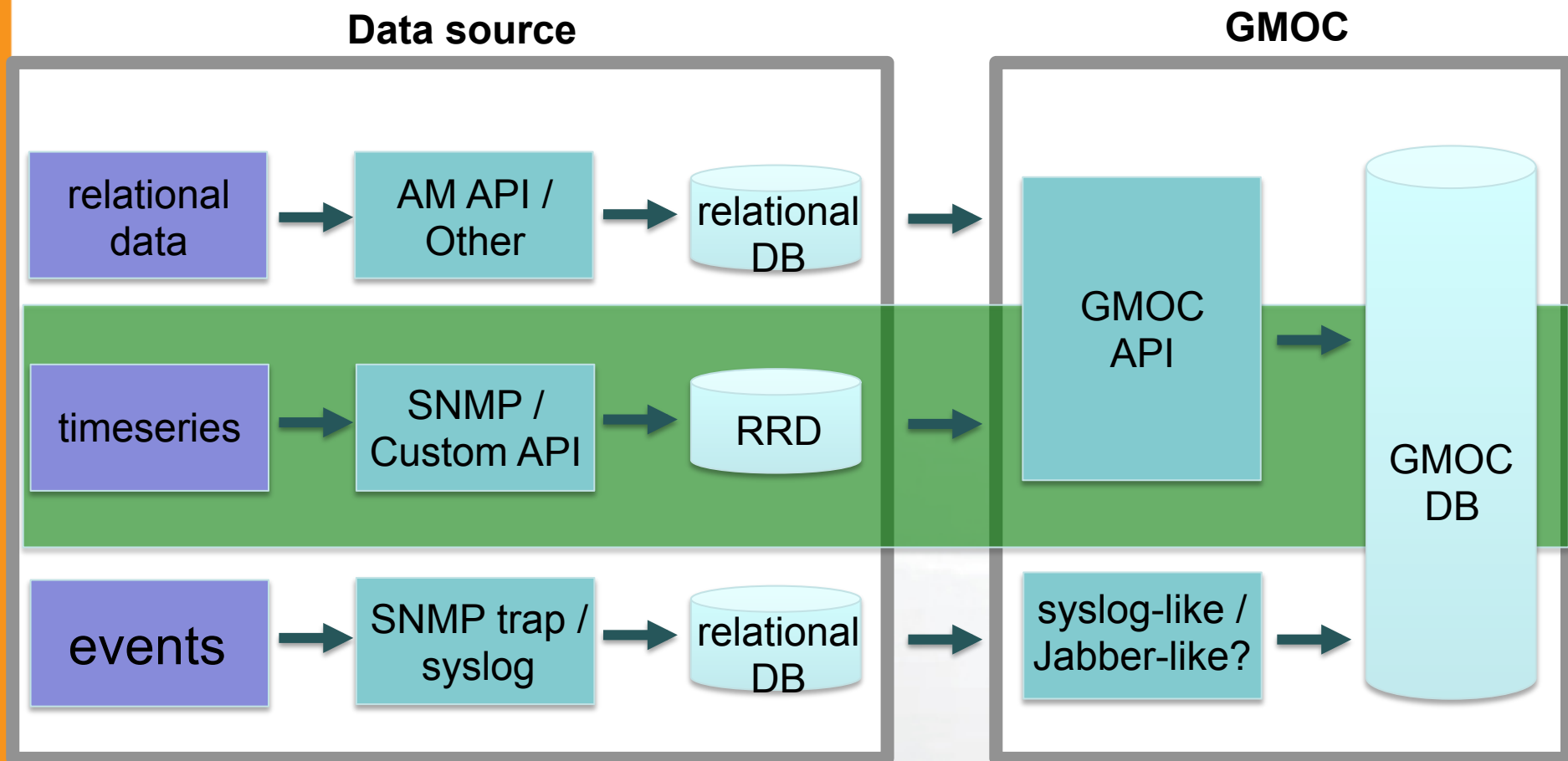
3) Building adaptable tools?

Publishing Data



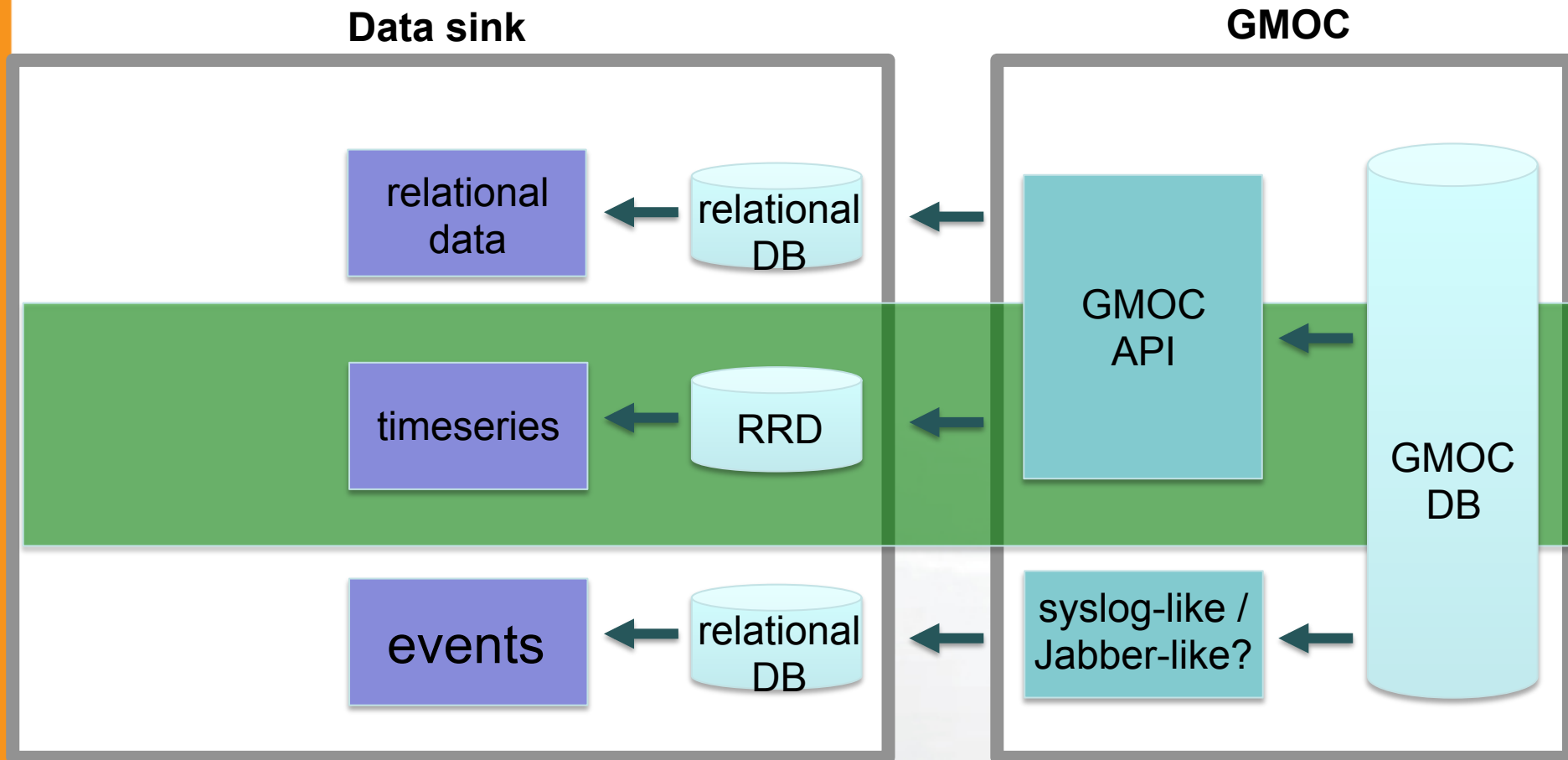
3) Building adaptable tools?

Publishing Data






3) Building adaptable tools?

Accessing Data



4) What's next?

Status		Requirement/Pain Point
	(11)	
	(12)	
	(13)	
	(14)	
	(15)	
	(16)	
	(17)	
	(18)	
	(19)	
	(20)	
	(21)	Others?