FuncX Architecture Update

Yadu Babuji (yadu@uchicago.edu)

Parsl & funcX Fest 2022

What's new?

Major Update **v1.0.x** is now in production (Aug 16th 2022)

- New `FuncXExecutor` interface

- Architectural updates to funcX hosted services

- `funcx-endpoint` updates

- Better security

What's new? FuncXExecutor

Simplifies interacting with funcX

```python
hello_funcx.py                                                    Raw

1    from funcx import FuncXClient, FuncXExecutor
2
3    fx = FuncXExecutor(FuncXClient())
4
5    def remote_hello():
6        import platform
7        remote_node_name = platform.uname().node
8        return "Hello from {}".format(remote_node_name)
9
10   tutorial_endpoint = '4b116d3c-1703-4f8f-9f6f-39921e5864df' # Public tutorial endpoint
11
12   future = fx.submit(remote_hello, endpoint_id=tutorial_endpoint)
13
14   print(future.result()) # Blocking wait until result
```
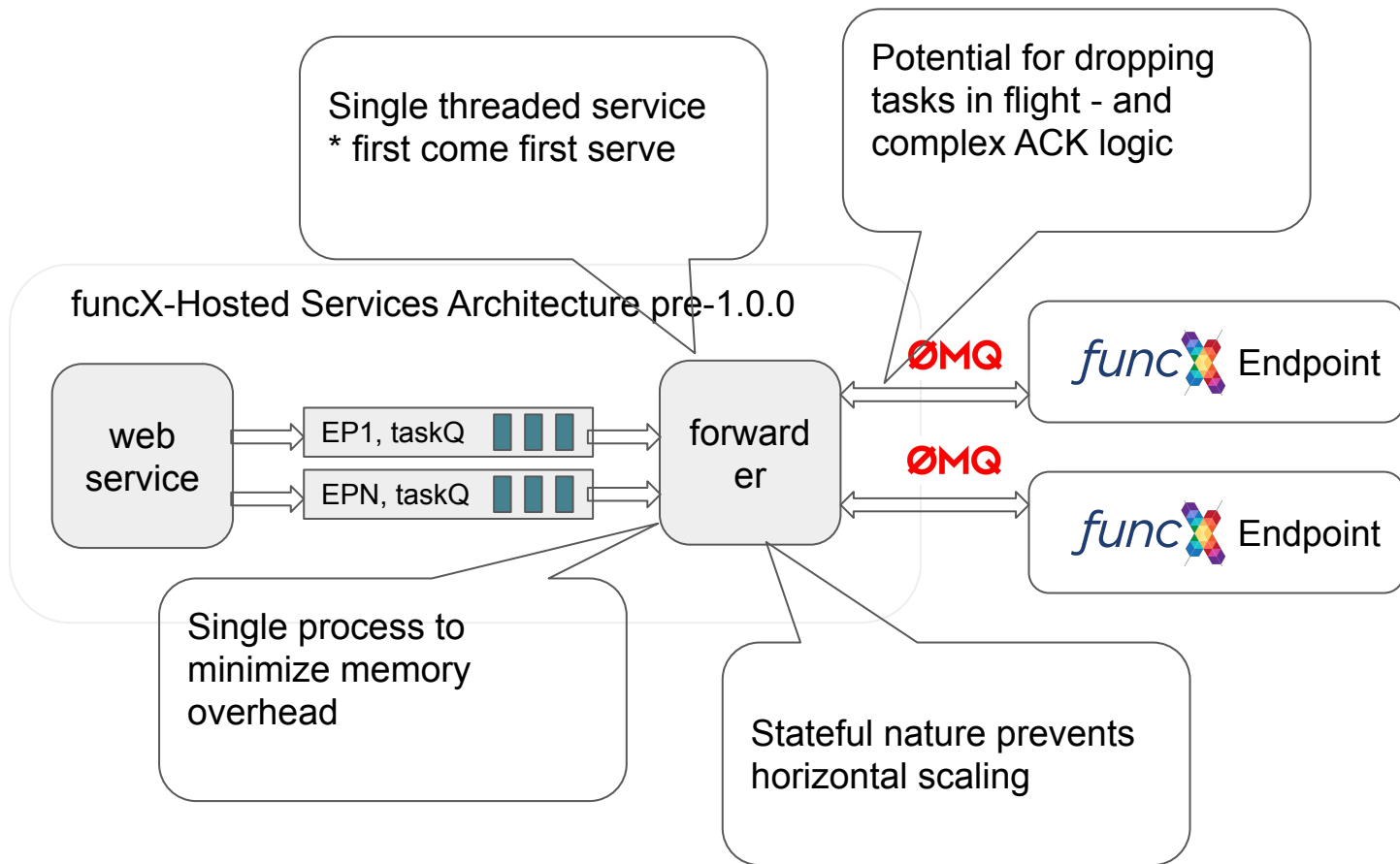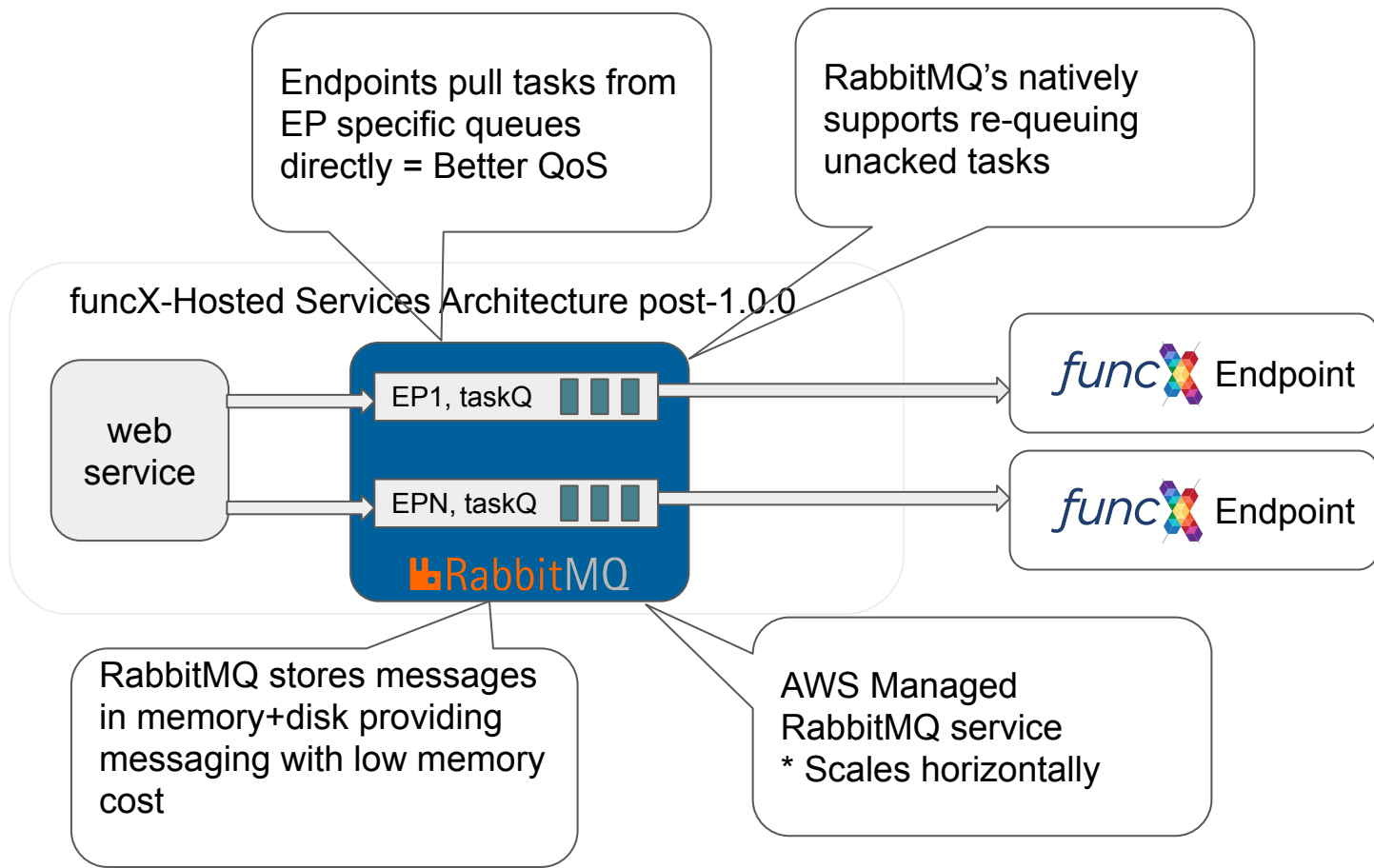
What's new? funcX Service Architecture

1. The previous architecture worked well for a few users actively using funcX, while we wanted ~1000s of users to be able to actively use the system
   a. Ensure **Quality of Service (QoS)** with a mix of bursty workloads and high-throughput workloads
   b. **Memory efficiency** when handle task backlogs when endpoints are offline
   c. Enable **horizontal-scaling** to handle varying workloads
   d. Guarantee that we don't drop **tasks**
2. **Low-latency** result delivery
3. As funcX matures we need better **security** for the pipes

Architecture Updates: Scaling to +1000s of users

Single threaded service
* first come first serve

Potential for dropping tasks in flight - and complex ACK logic

funcX-Hosted Services Architecture pre-1.0.0

web service

EP1, taskQ

EPN, taskQ

forwarder

ØMQ

ØMQ

funcX Endpoint

funcX Endpoint

Single process to minimize memory overhead

Stateful nature prevents horizontal scaling

Architecture Updates: Scaling to +1000s of users

Endpoints pull tasks from EP specific queues directly = Better QoS

RabbitMQ's natively supports re-queuing unacked tasks

funcX-Hosted Services Architecture post-1.0.0

web service

EP1, taskQ

EPN, taskQ

RabbitMQ

*func*X Endpoint

*func*X Endpoint

RabbitMQ stores messages in memory+disk providing messaging with low memory cost

AWS Managed RabbitMQ service
* Scales horizontally

Architecture Updates: Low latency result delivery
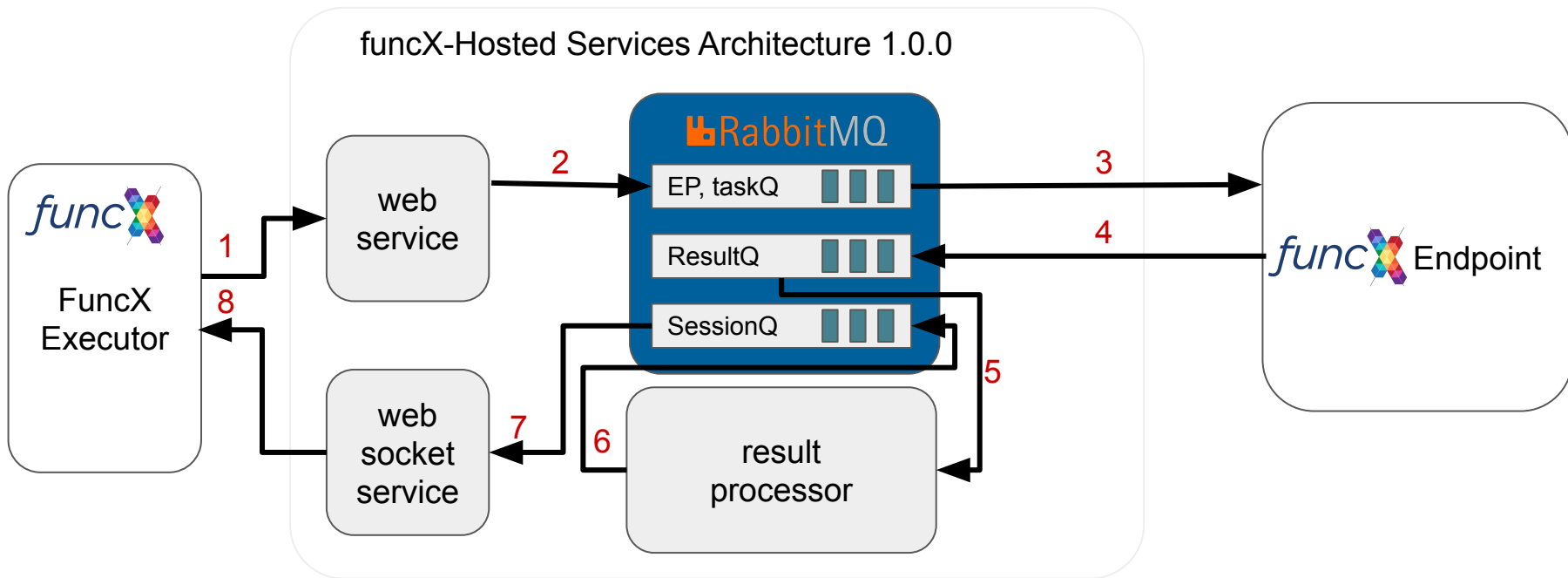
What's new? Better security

funcX-Hosted Services Architecture post-1.0.0

EP specific TaskQ can only be read from with user creds minted at EP registration

EPN, taskQ

EPN, ResultsTopic

ResultsQ

RabbitMQ

web service

2. Reset user creds

EP specific topic for results is keyed to user-creds

3. Return user credentials

4. Connect with new creds over AMQPS

1. Register EP
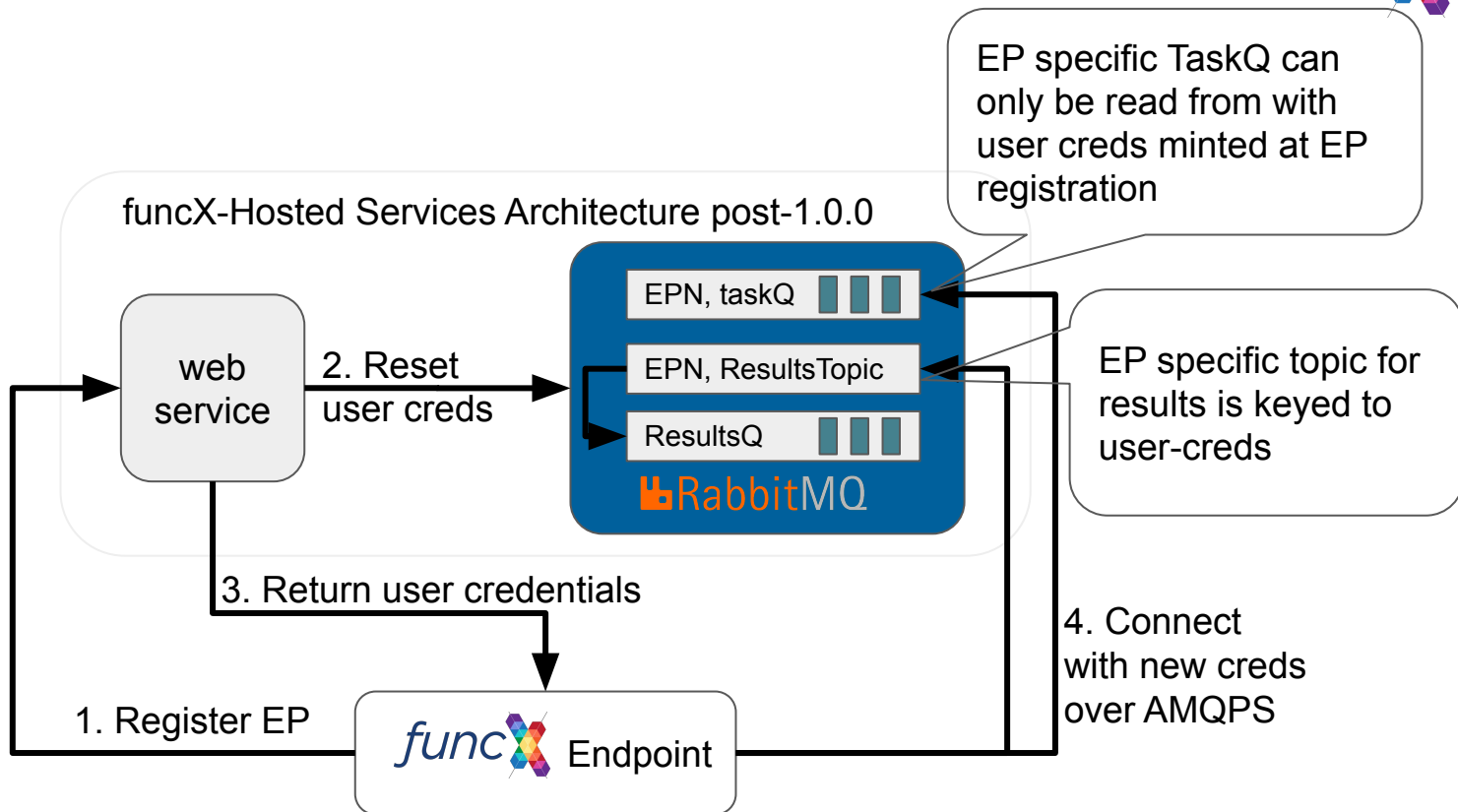
funcX Endpoint

What's new? funcx-endpoint

- Updated wire protocol to communicate with hosted services

- funcx-endpoint now uses RabbitMQ instead of ZeroMQ for persistent connections:

  - Outbound ports 5671 (AMQPS) and 443 are now required

- Fresh credentials are minted on every registration

Fin

Questions?

Slack and Email info are here -> https://funcx.org/support.html