

Redis 7 and Beyond

Meir Shpilraien, Yossi Gottlieb
& Itamar Haber



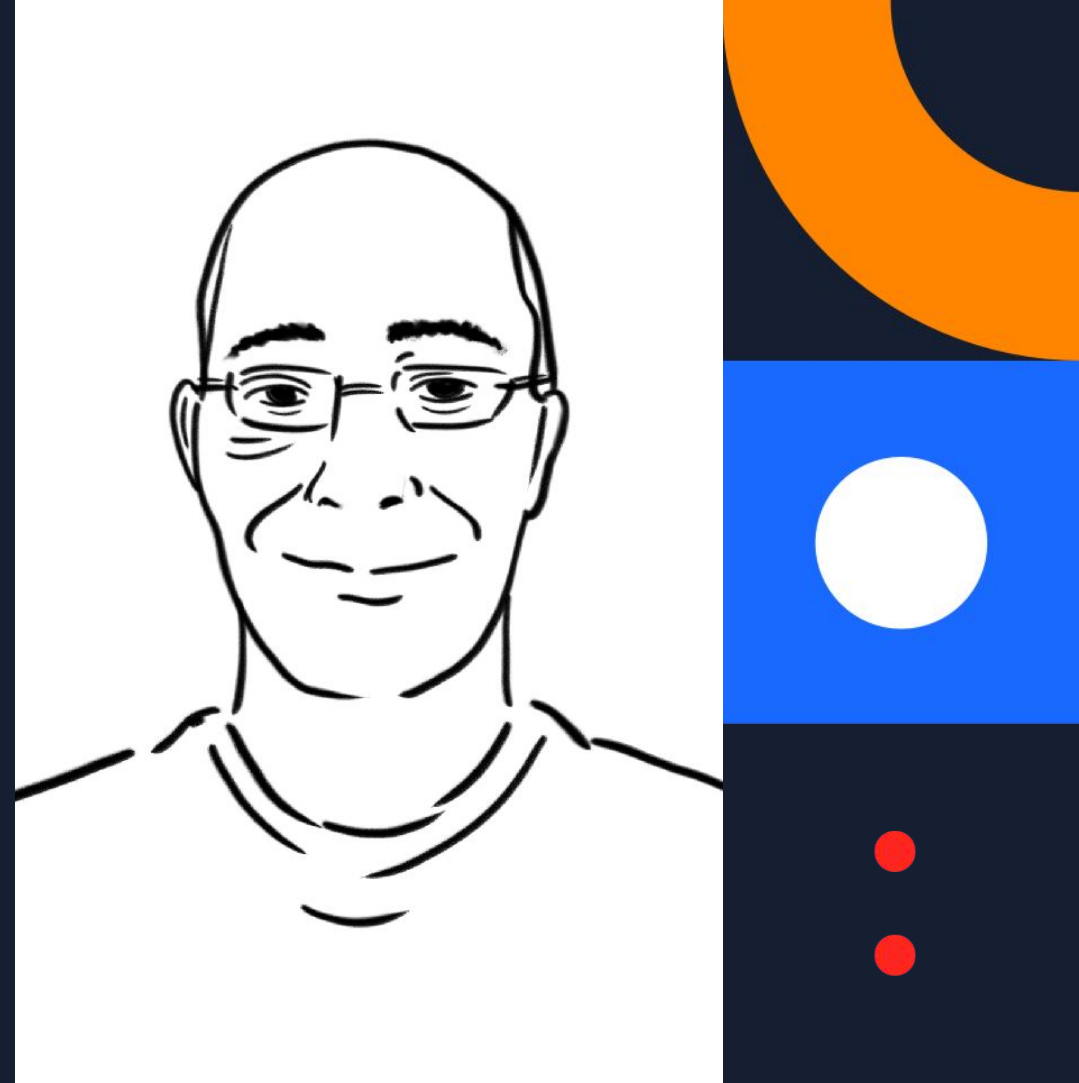
Itamar Haber

Technology Evangelist @ Redis Labs

I'm a Redis Geek
Community Leader @ Redis Core Team

I love counting: one, two, three! Ah ah ah!
I have an obsession with data, technology and people around it

Partner of the One, fathered the Three & adoring my dog



redisconf2021

Redis 7: Agenda

1. Last year in Open Source Redis
2. Redis Functions
3. RedisRaft
4. A Glimpse into the Future



redisconf2021

Redis 7 in a Nutshell

- To be released later this year
- To be driven and developed by the community
- An opportunity to introduce new features, major changes & break stuff
- Some features are already being designed and/or implemented
- Others are waiting to be...

Last Year in Open Source Redis

- Apr 30, 2020: Redis 6.0
- Jun 30, 2020: Antirez steps back
- Jul 1, 2020: New project governance model
- Jul 9, 2020: Redis 6.0.6
- Feb 22, 2021: Redis 6.2 "Community Edition"
- Feb 26, 2021: 🎂 Redis is 12 years old

Some of the changes:

- Core team made up of 5 members
- 14 Redis Contributors appointed
- A revamped release process
- Open discussions & plans
- Pull Request review process

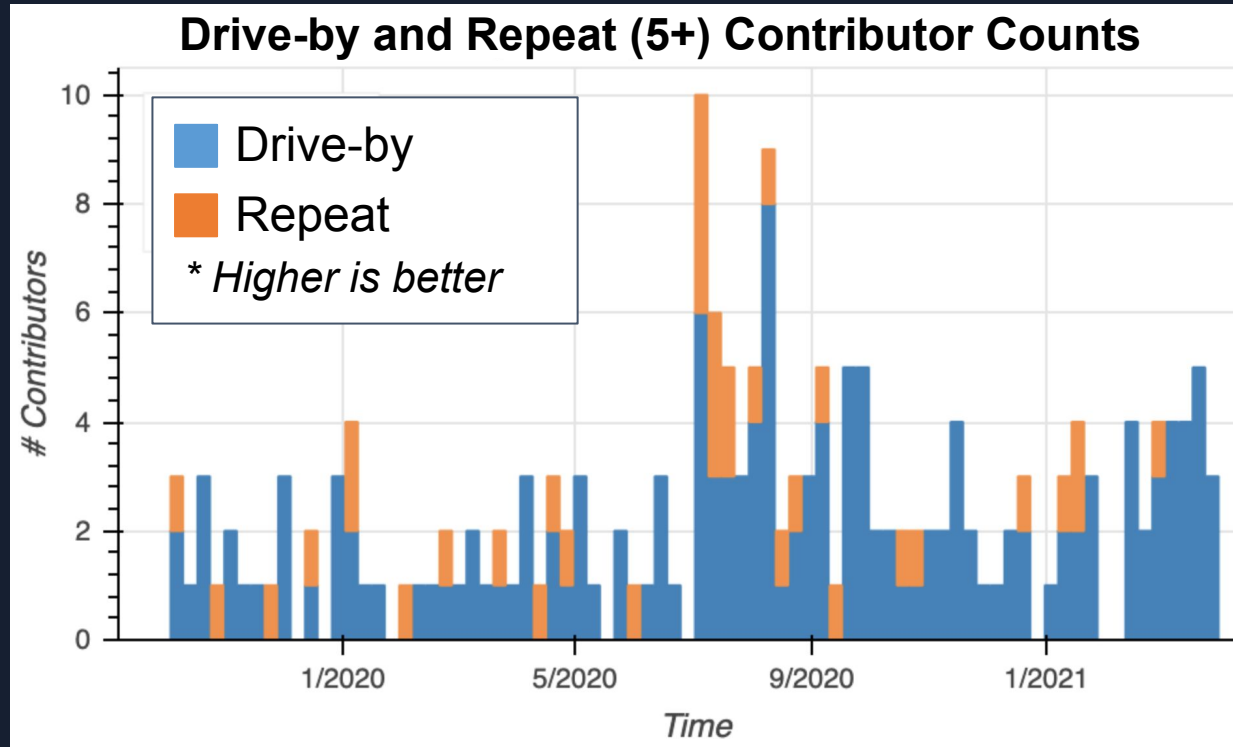


redisconf2021

The governance model is intended to be a meritocracy, aiming to **empower individuals** who demonstrate a long-term commitment and make significant contributions



Code Contributors



Data from the unstable branch:

- **Before:** 9 months between 2019-10-01 and 2020-06-30
- **After:** 9 months between 2020-07-01 and 2021-03-31

Metric	Before	After	Change
PRs Created	526	820	56%
PRs Closed	421	1078	156%
PRs Open	889	634	-29%
Git Authors	80	149	86%

Brewed in cauldron.io from  Bitergia



redisconf2021

Redis 7 and Beyond: The Content



redisconf2021

Redis Functions

Meir Shpilraien



Agenda

Lua

Current approach to scripting in Redis

Modules

Powerful but less accessible

RedisGears

A Redis Module, more powerful than Lua and less complex than module

Redis Function

New approach to programmability in Redis



redisconf2021

Lua Scripting

- Added in Redis 2.8
- Allows compose logic that runs server side
- Considered part of the client application, As such:
 - Not replicated
 - Not persisted
 - May be lost at any time
- Client application must keep the Lua scripts in the application itself, and handle versioning and compatibility across clients



Redis Modules

- Added in version 4.0
- Can extend Redis in an almost any way
- Potentially unsafe (a bug in module will crash the Redis)
- Written in C (or some other low level language), which is less preferred by most developers
- Not part of Redis core



redisconf2021

RedisGears

- A Redis module (disclaimer: author here)
- Provide Python API (Java, JS, C, and more languages are coming)
- Cluster agnostic
- Supports Event processing



redisconf2021

Redis Functions - Proposal

- Enhance the existing Lua scripts support. As such, functions have:
 - Full interaction with Redis using commands invocation
 - Atomicity
- Considered part of the server. As such, Functions are:
 - Replicated
 - Persisted
 - Named
- Invocation is done by the function name
- Agnostic to the programming language (Lua5.1 and QuickJS)



redisconf2021

Redis Functions API

- FUNCTION CREATE ENGINE_NAME [REPLACE] [ARGS_DESCRIPTOR <ARGS_DESCRIPTOR>] [DESC <DESCRIPTION>] BLOB
- FUNCTION CALL NAME NUM_KEYS key1 key2 ... ARGS arg1 arg2
- FUNCTION INFO NAME
- FUNCTION DELETE NAME



redisconf2021

Demo



redisconf2021

RedisRaft in a Nutshell

- Strong consistency (strict serializable) Redis deployment option
- 100% Redis Module
- Open-source dual licensing (AGPLv3/RSAL)

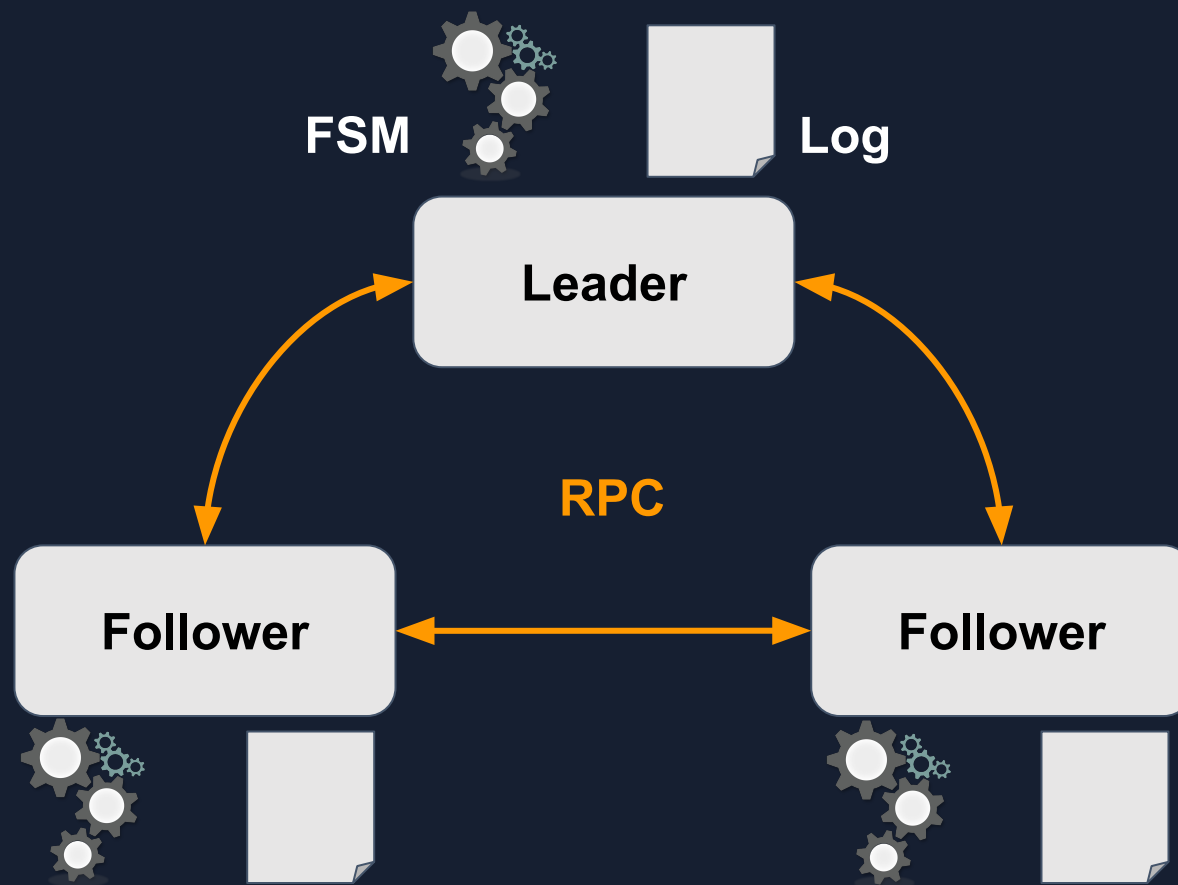


redisconf2021

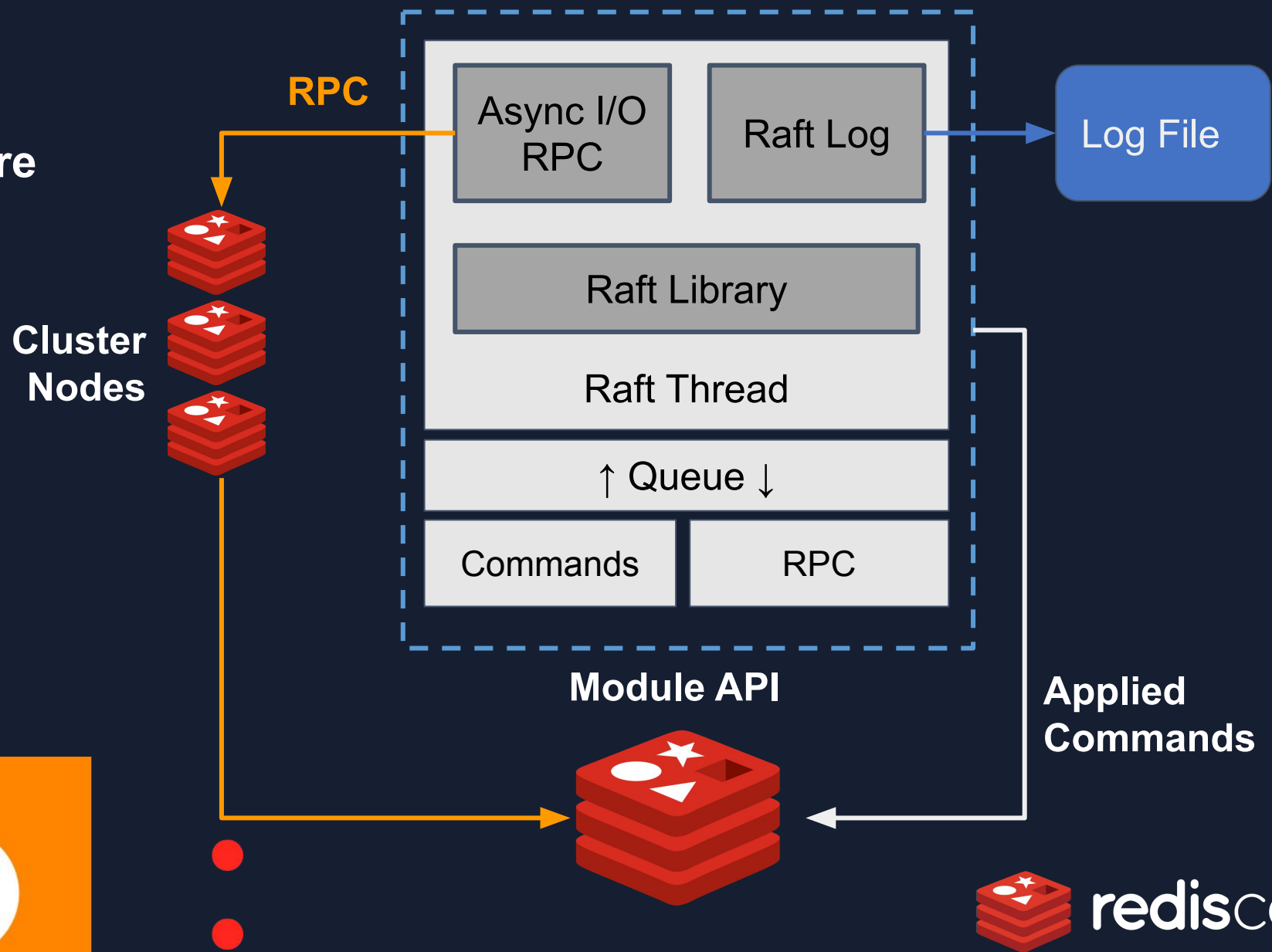
Raft in a Nutshell

- Leader Election
- Log Replication

• Commit & Apply to FSM
For more information, see:
<https://raft.github.io>



Architecture



redisconf2021

`/bin/bash`
\$



redisconf2021

Where we stand?

- More work to do before 1.0
- Jepsen testing round #1 (Q1/2020)
- First release - H2/2021 🙌

<https://github.com/RedisLabs/RedisRaft>



redisconf2021

RedisRaft \in Redis 7

- Our mission is scaling up
- Redis becomes an ecosystem
 - Clients
 - Modules
 - Dev & Ops Tools



redisconf2021

Redis 7 and Beyond: The Known Knowns...



redisconf2021

Science Fiction, a.k.a Projects/4

- Nested Element Expiry
- Reliable Keyspace Notifications
- Key Annotations
- Cluster & Sentinel: next steps
- Incremental evolutions: ACL, Streams, multi-part AOF, Module API, ...
- Place your bets?



redisconf2021

History Will Judge Us

- It is hardly only about the code
- It is a collaborative effort by a **community**
- There's more than one way to peel this banana, including #HelpWanted: adding your opinion, giving feedback, designing features, reviewing code, testing the implementation, documenting everything, triaging new issues, reproducing known bugs, optimizing performance, solving hard problems...
- See you on github.com/redis/redis



redisconf2021

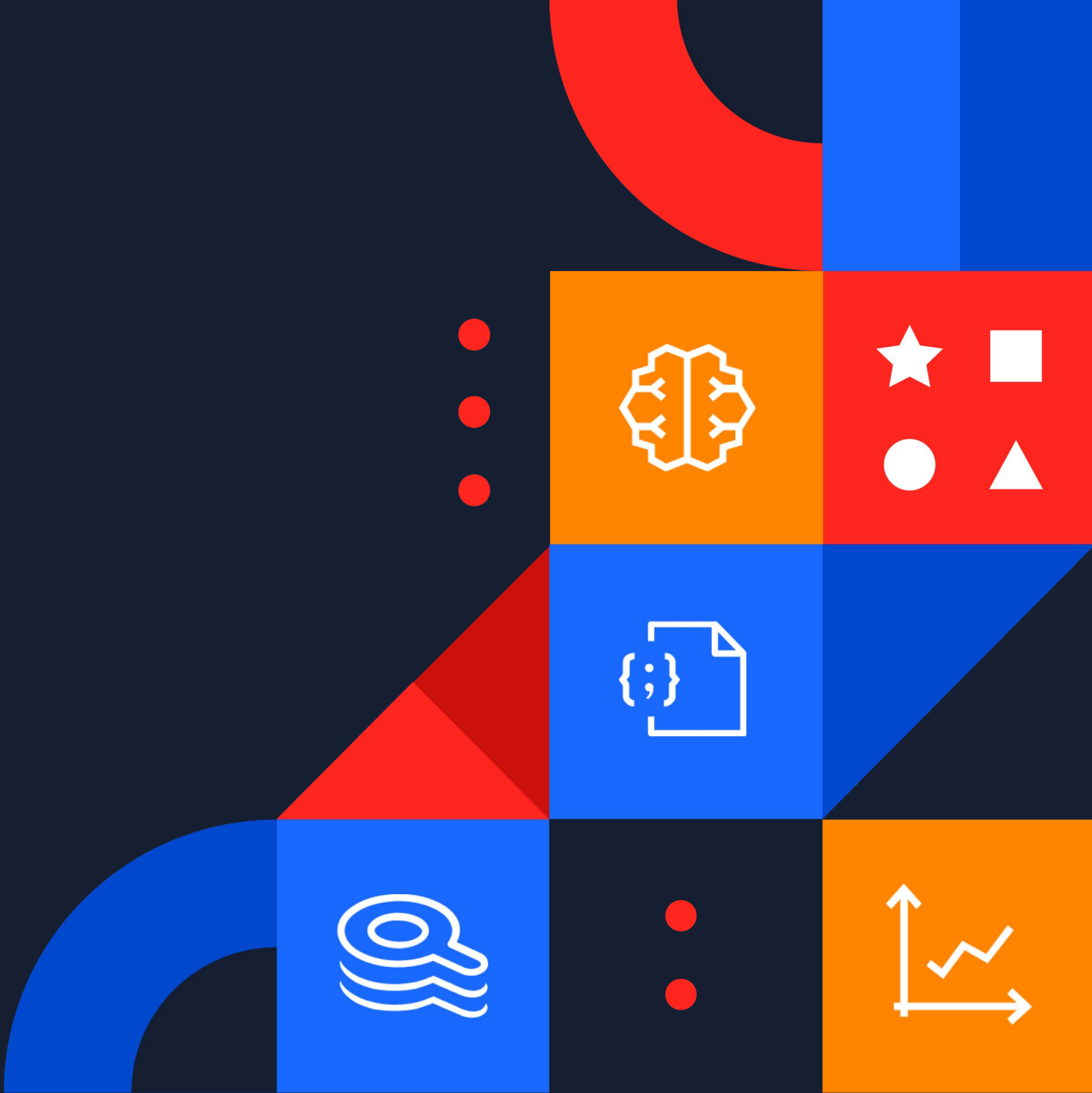
Thank you.

itamar@redislabs.com

www.redislabs.com



redisconf2021



Redis Functions

Meir Shpilraien



Agenda

Lua

Current approach to scripting in Redis

Modules

Powerful but less accessible

RedisGears

A Redis Module, more powerful than Lua and less complex than module

Redis Function

New approach to programmability in Redis



redisconf2021

Lua Scripting

- Added in Redis 2.8
- Allows compose logic that runs server side
- Considered part of the client application, As such:
 - Not replicated
 - Not persisted
 - May be lost at any time
- Client application must keep the Lua scripts in the application itself, and handle versioning and compatibility across clients



redisconf2021

Redis Modules

- Added in version 4.0
- Can extend Redis in an almost any way
- Potentially unsafe (a bug in module will crash the Redis)
- Written in C (or some other low level language), which is less preferred by most developers
- Not part of Redis core



redisconf2021

RedisGears

- A Redis module (disclaimer: author here)
- Provide Python API (Java, JS, C, and more languages are coming)
- Cluster agnostic
- Supports Event processing



redisconf2021

Redis Functions - Proposal

- Enhance the existing Lua scripts support. As such, functions have:
 - Full interaction with Redis using commands invocation
 - Atomicity
- Considered part of the server. As such, Functions are:
 - Replicated
 - Persisted
 - Named
- Invocation is done by the function name
- Agnostic to the programming language (Lua5.1 and QuickJS)



redisconf2021

Redis Functions API

- FUNCTION CREATE ENGINE_NAME [REPLACE] [ARGS_DESCRIPTOR <ARGS_DESCRIPTOR>] [DESC <DESCRIPTION>] BLOB
- FUNCTION CALL NAME NUM_KEYS key1 key2 ... ARGS arg1 arg2
- FUNCTION INFO NAME
- FUNCTION DELETE NAME



redisconf2021

Demo



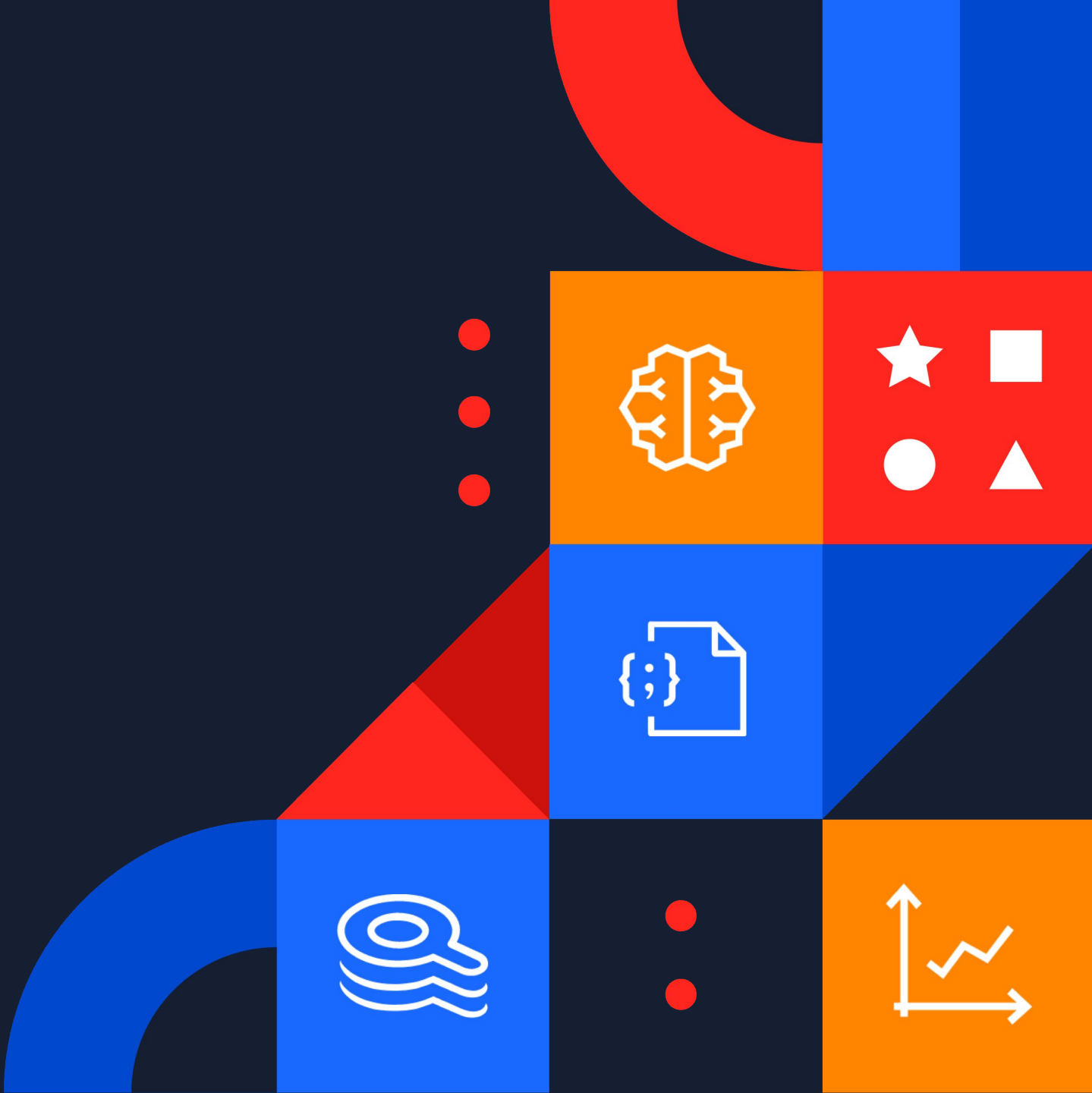
redisconf2021

Thank you.

meir@redislabs.com



redisconf2021



RedisRaft in a Nutshell

- Strong consistency (strict serializable) Redis deployment option
- 100% Redis Module
- Open-source dual licensing (AGPLv3/RSAL)

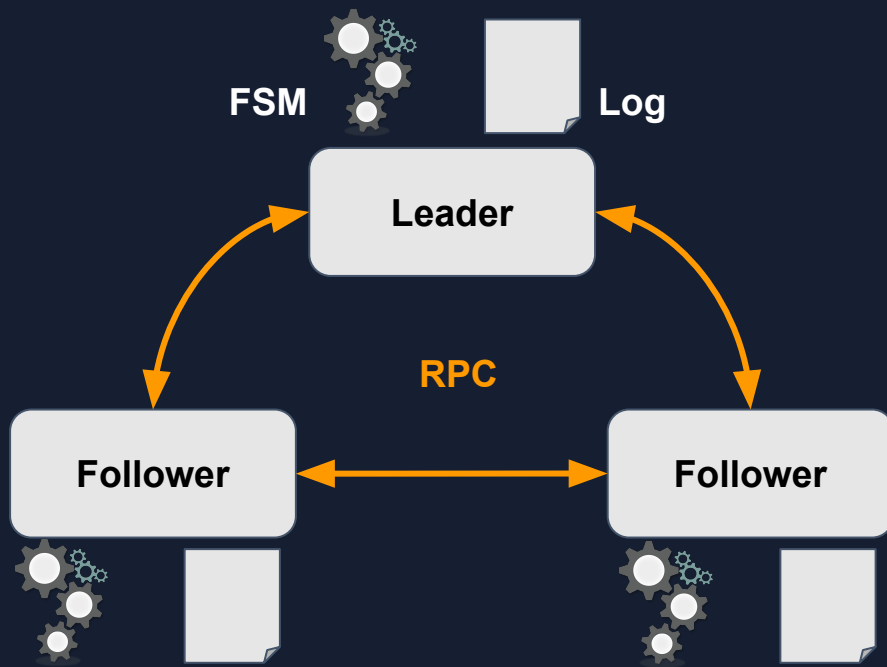


redisconf2021

Raft in a Nutshell

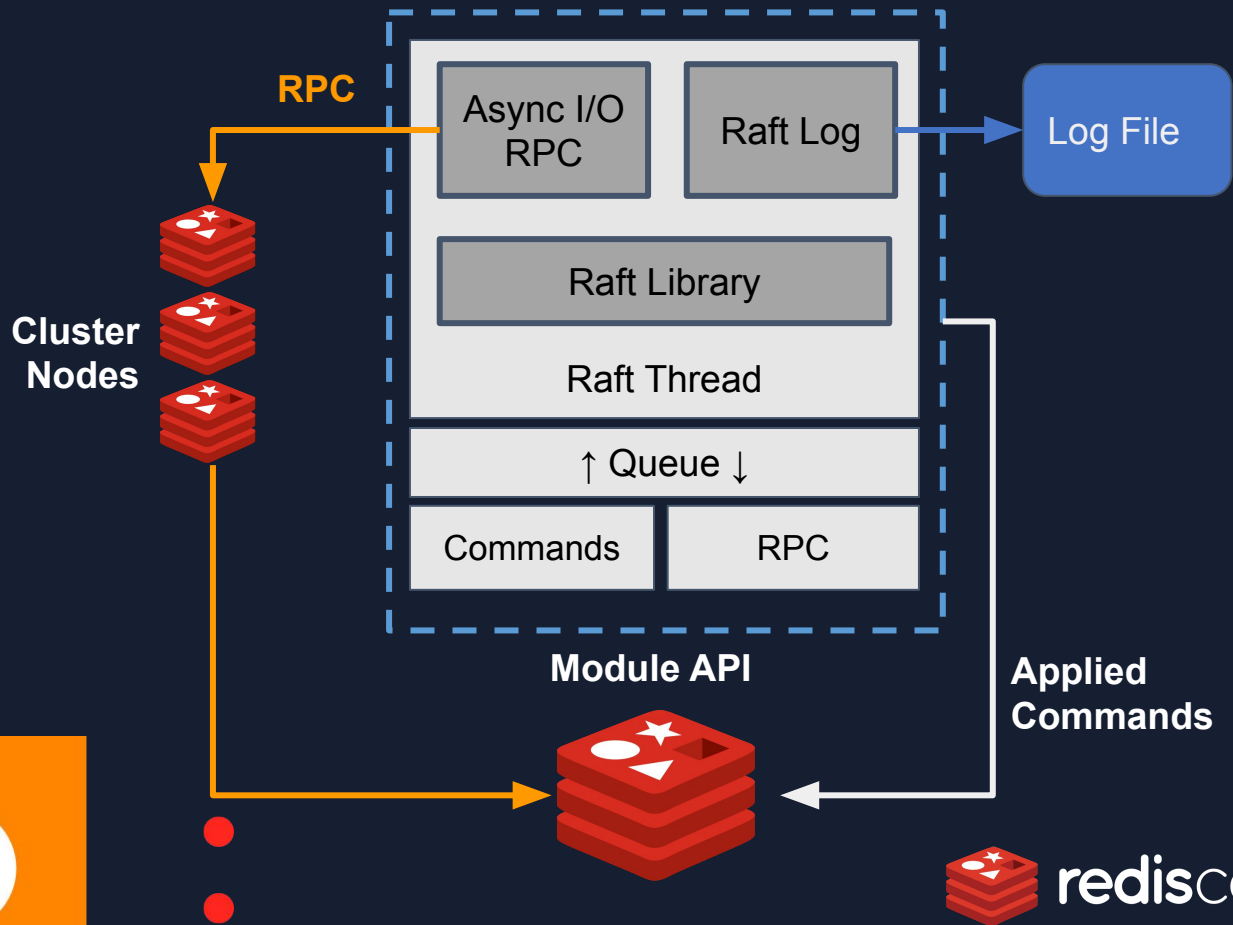
- Leader Election
- Log Replication
- Commit & Apply to FSM

For more information, see:
<https://raft.github.io>



redisconf2021

Architecture



redisconf2021

```
/bin/bash  
$
```



redisconf2021

Where we stand?

- More work to do before 1.0
- Jepsen testing round #1 (Q1/2020)
- First release - H2/2021 🍀

<https://github.com/RedisLabs/RedisRaft>



redisconf2021

RedisRaft \subseteq Redis 7

- Our mission is scaling up
- Redis becomes an ecosystem
 - Clients
 - Modules
 - Dev & Ops Tools



redisconf2021

Redis 7 and Beyond: The Known Knowns...



redisconf2021

Science Fiction, a.k.a Projects/4

- Nested Element Expiry
- Reliable Keyspace Notifications
- Key Annotations
- Cluster & Sentinel: next steps
- Incremental evolutions: ACL, Streams, multi-part AOF, Module API, ...
- Place your bets?



redisconf2021

History Will Judge Us

- It is hardly only about the code
- It is a collaborative effort by a **community**
- There's more than one way to peel this banana, including #HelpWanted: adding your opinion, giving feedback, designing features, reviewing code, testing the implementation, documenting everything, triaging new issues, reproducing known bugs, optimizing performance, solving hard problems...
- See you on github.com/redis/redis



redisconf2021

Thank you.

itamar@redislabs.com

www.redislabs.com

