

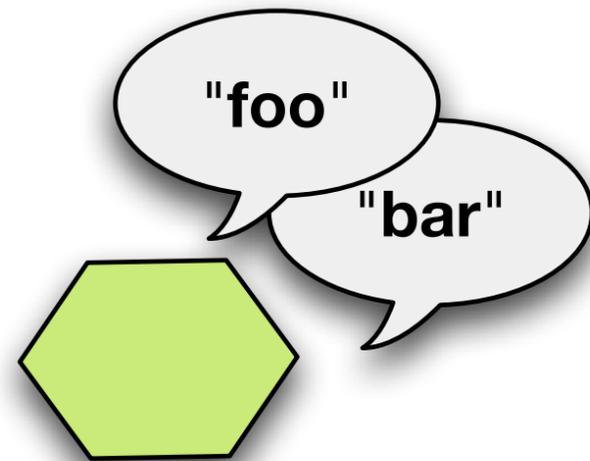
AMQP at  [idealists.org](https://www.idealists.org)

Basic Queueing

- Push message “ABC” onto queue Q
- Consume a message from queue Q
- Atomic operations

Basic Publish/Subscribe

note lack of connecting lines



AMQP

- Yet another messaging standard
- Can be thought of as only offering publish/subscribe
- Has sufficient knobs to implement simple queueing and other patterns
- pub/sub at heart is actually pretty great in practice*

AMQP

- Standard is machine-readable XML meant to be mechanically converted into driver implementations
- YES!!!! I I ! I !!!

AMQP

- Protocol is meh but fine
- Can express interesting messaging patterns

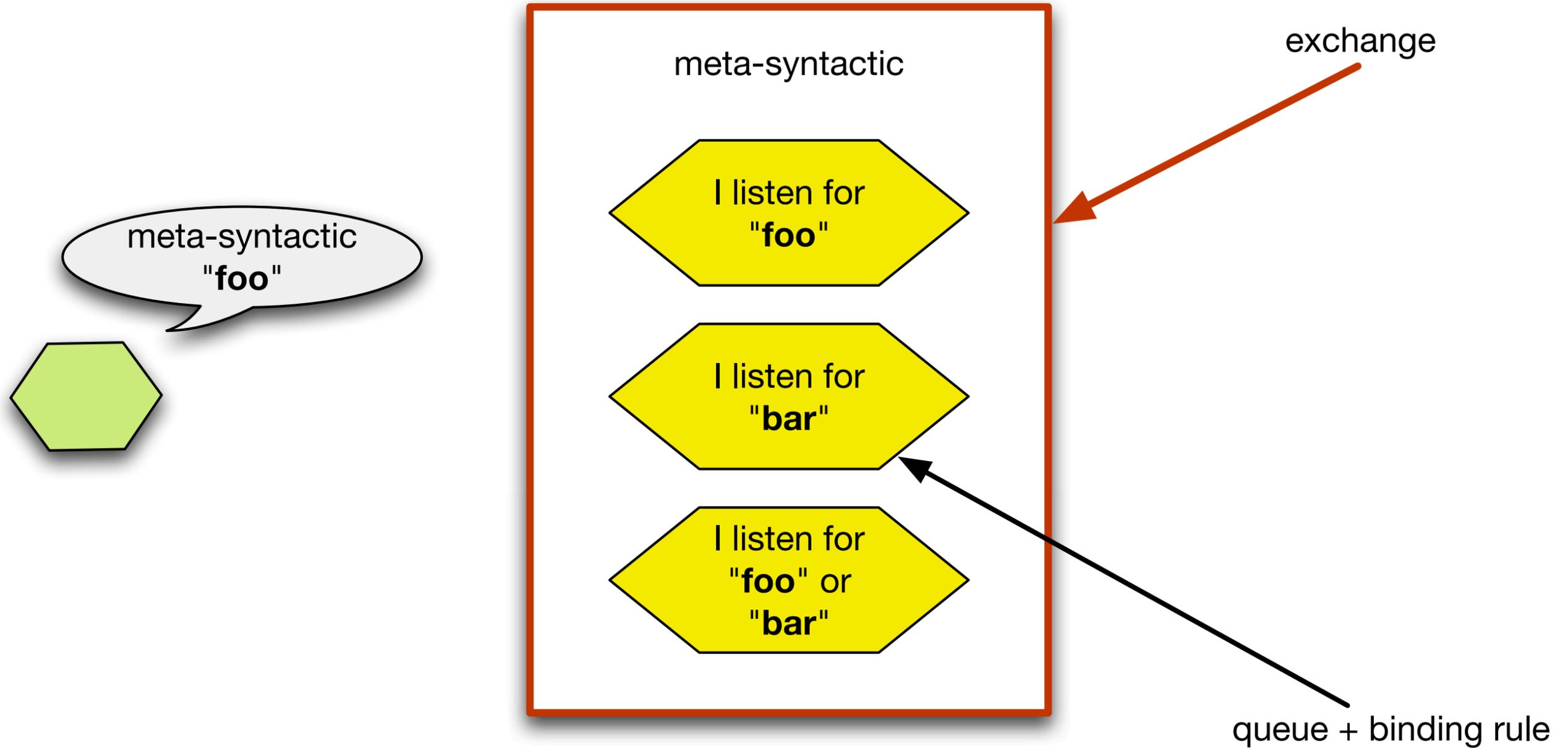
AMQP

Double-labeled Messages?

exchange Subject:
 ==
topic Subject2:

Exchanges are typed and provide different enqueueing behavior
Some types consider topic, some don't

AMQP



RabbitMQ

An Erlang AMQP broker implementation

- Well-known for
 - Speed
 - Being widely deployed
 - Somehow awesome because it's *Erlang*!

At Idealist

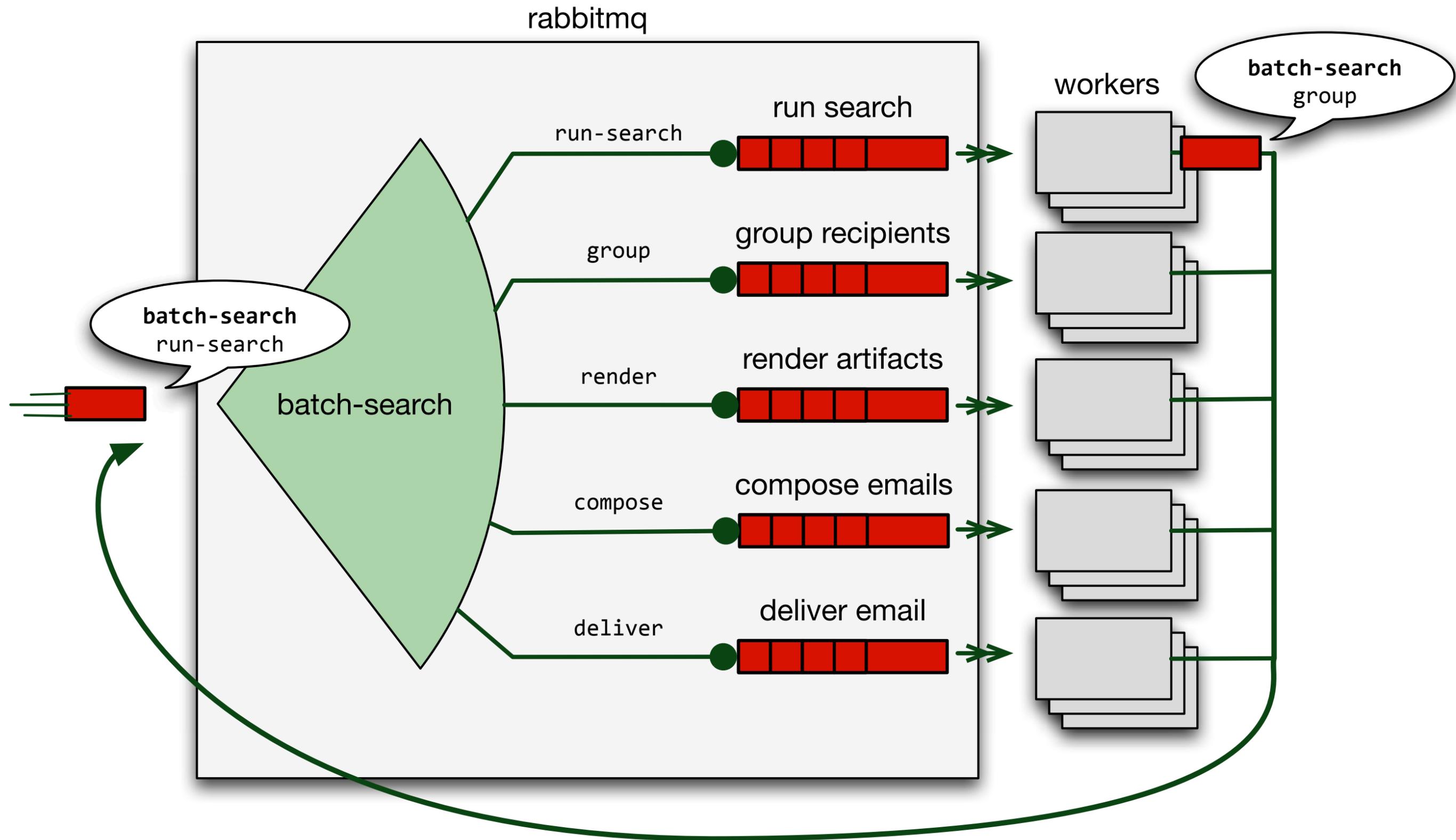
- Traditional web cluster design: balancers, app servers, data stores, redundancy
- Generally synchronous communication between user-facing apps and inside services during app requests (e.g. SQL / noSQL reads block)
- Most complex request side-effects are kicked off asynchronously with AMQP via loosely coupled observers
- Since it's there, we use AMQP for general processing and message bus purposes too

Snowball

A simple scaling pipeline

- Users can receive search results periodically by email
- Many users are interested in the same thing
“Part-time volunteer opportunities in New York City”
- Some of those people differ in their preferred language or other characteristics that affect messaging
- Not duplicating processing effort would be swell

- Run a search, retrieve results → 1 message
- Segment recipients by language, etc. → 1-8 messages
- Render search results in segment-specific ways → 1-30,000 messages
- Compose individualized email per recipient → 1 message
- Send SMTP → 1 message



Queue	Message Size	Fill	Drain	Typical Q Size
run search 	50 bytes	6000/sec	500/sec	150,000
group recipients 	1k	-	3000/sec	4
render artifacts 	1k	-	2000/sec	2
compose emails 	16k	-	500/sec	12,000
deliver email 	100k	-	1000/sec	10

numbers loosely inspired by actual facts

CRUD

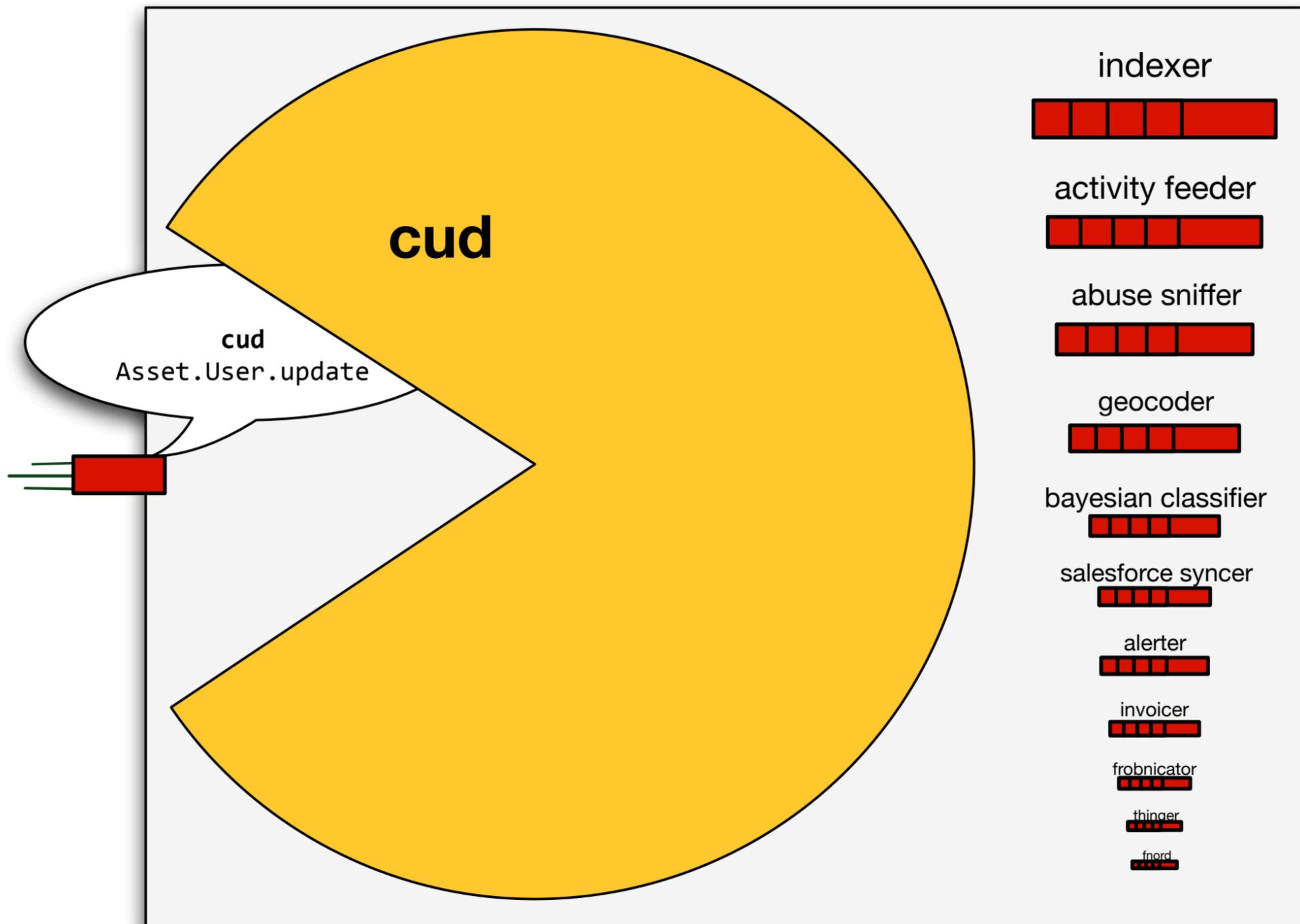
Observing change

At Idealist

- Reads are frequent and dirt cheap
- Uninteresting writes are infrequent and kinda cheap (e.g. updating statistics)
- Interesting writes are rare and relatively expensive (e.g. burst of SQL DML)
- **So why not make them REALLY expensive?**

- Variety of domain objects or documents; many with deep structure
- Canonically stored in SQL, also capable of dirt simple reversible JSON key/value serialization
- Domain object creation, update and deletion (**CUD**) tracking can be enabled for any process

- CUD events emit MQ messages to the cud exchange
- Message contains enough state diff information to reconstitute the full before and after states of the domain object
- Message contains light context (whodunnit)



- **Low volume: <100/sec Hefty: ~6k Wide: ~40 subscriptions**
- Decoupled behavior tied to state change
- Diff transmission eliminates DB I/O for some classes of consumers
- Any process can emit messages to cud, including cud consumers
- All cud consumers also provide a synchronous API for testing (yay) and clever (boo) use cases

pub/sub at heart is actually pretty
great in practice*

- The inability to directly delivery to a queue has pleasant side effects
- Make copies of messages in new queues as an afterthought
- If you've ever used `tee`, or had MySQL's insane(ly) flexible replication got you out of a production jam (oh ****, we need this data over there too, now!)
- Tremendously useful for migrating running services consuming active streams
- But.

- AFAIK, can't change the type of an exchange while it's running
- Sometimes service migration also involves a new version of your message format
- Adding more queues doesn't directly help that

At Idealist

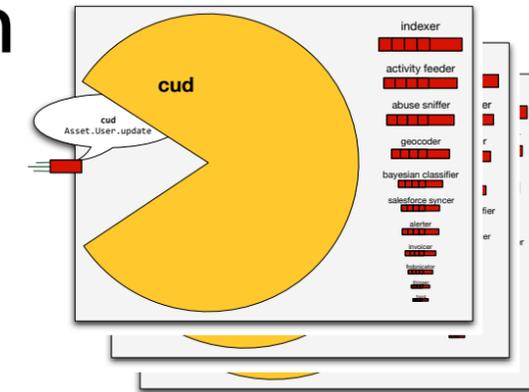
- Our production cluster can host n parallel running configurations
- Though only one is routing traffic for `.org`
- Maintaining backward compatibility for message formats, especially in `cod`, is not a lot of fun

Virtual Hosts

- AMQP has a notion of a “virtual host”, a private namespace for exchanges, queues, etc.

- Idealist makes a new virtual host per configuration

```
/config/56631  
/config/56632  
/config/...
```



- Only one moving part of configuration to twiddle; exchange + queue names are constant and easy to monitor

Thanks!

- Lots more stories!
- `jason@idealist.org` | `jek@discorporate.us`
jek on freenode
`@__jek__`