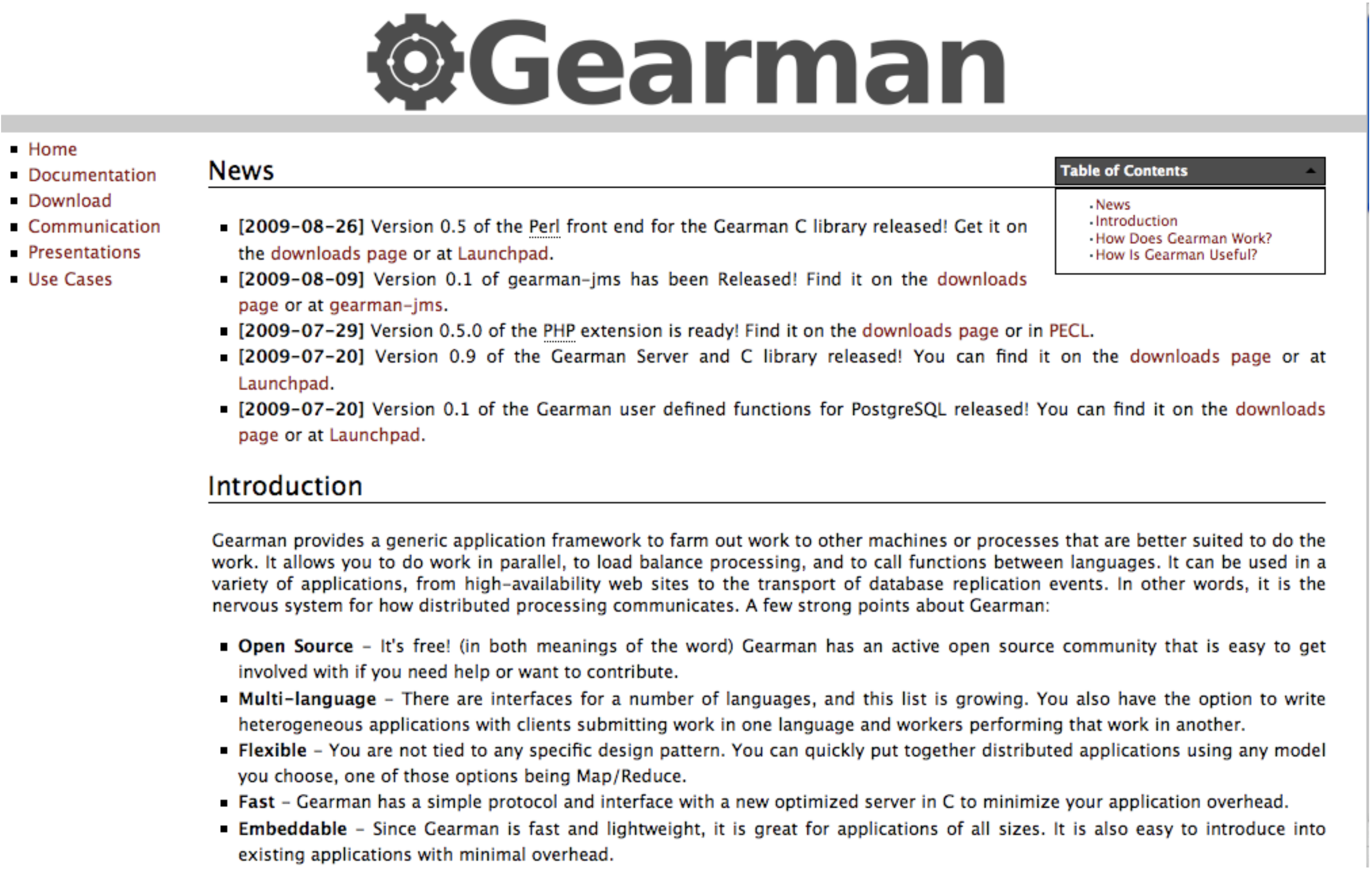
Gearman – working, later

WWBFD?



Gearman

- Home
- Documentation
- Download
- Communication
- Presentations
- Use Cases

News

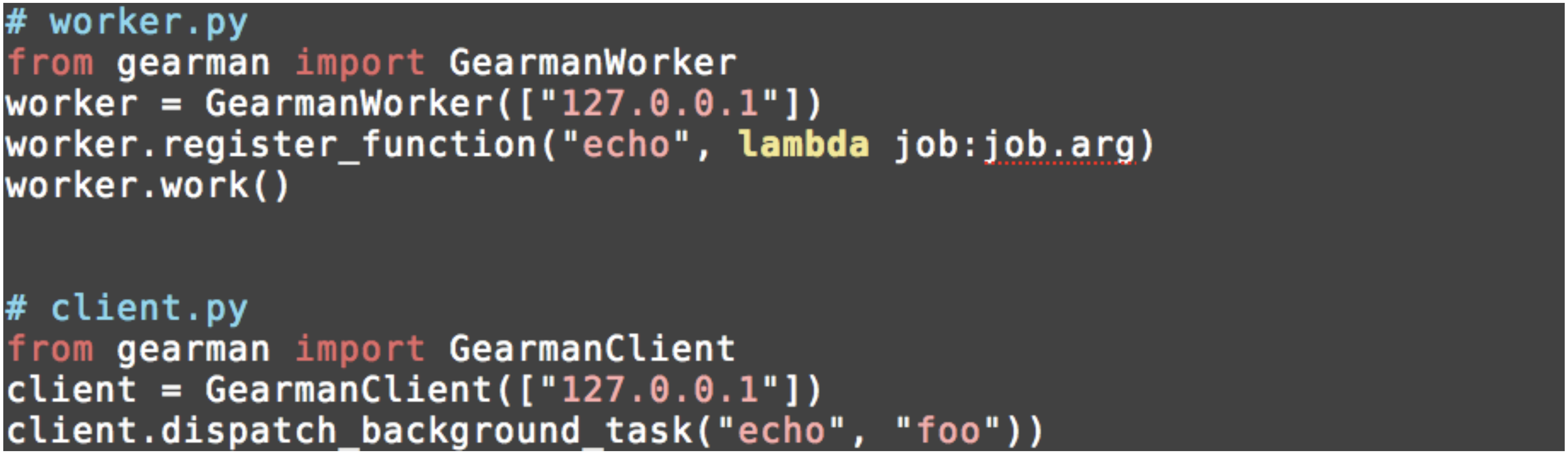
Table of Contents

- **[2009-08-26]** Version 0.5 of the Perl front end for the Gearman C library released! Get it on the downloads page or at Launchpad.
- **[2009-08-09]** Version 0.1 of gearman-jms has been Released! Find it on the downloads page or at gearman-jms.
- **[2009-07-29]** Version 0.5.0 of the PHP extension is ready! Find it on the downloads page or in PECL.
- **[2009-07-20]** Version 0.9 of the Gearman Server and C library released! You can find it on the downloads page or at Launchpad.
- **[2009-07-20]** Version 0.1 of the Gearman user defined functions for PostgreSQL released! You can find it on the downloads page or at Launchpad.

Introduction

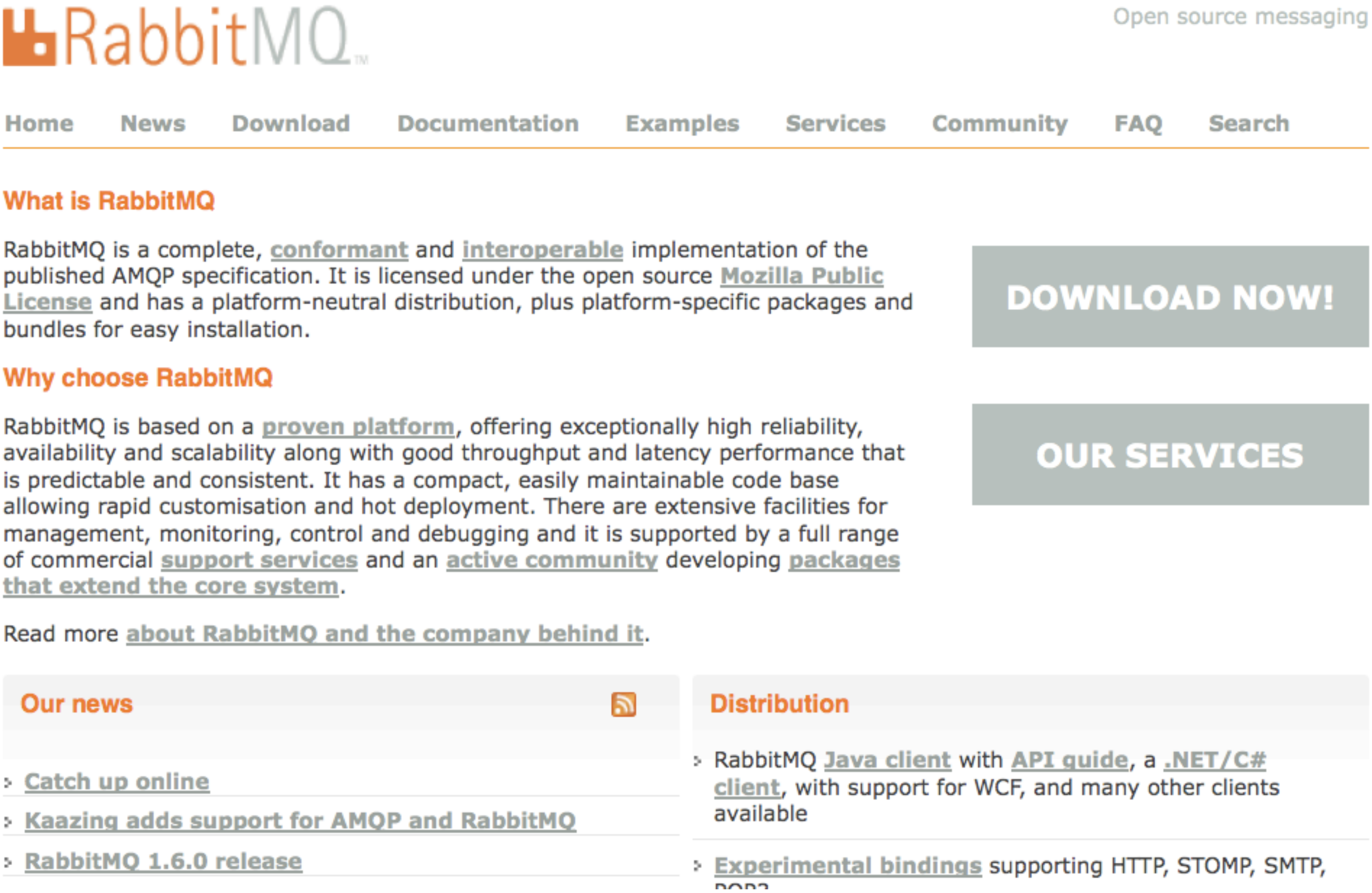
Gearman provides a generic application framework to farm out work to other machines or processes that are better suited to do the work. It allows you to do work in parallel, to load balance processing, and to call functions between languages. It can be used in a variety of applications, from high-availability web sites to the transport of database replication events. In other words, it is the nervous system for how distributed processing communicates. A few strong points about Gearman:

- **Open Source** – It's free! (in both meanings of the word) Gearman has an active open source community that is easy to get involved with if you need help or want to contribute.
- **Multi-language** – There are interfaces for a number of languages, and this list is growing. You also have the option to write heterogeneous applications with clients submitting work in one language and workers performing that work in another.
- **Flexible** – You are not tied to any specific design pattern. You can quickly put together distributed applications using any model you choose, one of those options being Map/Reduce.
- **Fast** – Gearman has a simple protocol and interface with a new optimized server in C to minimize your application overhead.
- **Embeddable** – Since Gearman is fast and lightweight, it is great for applications of all sizes. It is also easy to introduce into existing applications with minimal overhead.



```
# worker.py
from gearman import GearmanWorker
worker = GearmanWorker(["127.0.0.1"])
worker.register_function("echo", lambda job:job.arg)
worker.work()


# client.py
from gearman import GearmanClient
client = GearmanClient(["127.0.0.1"])
client.dispatch_background_task("echo", "foo"))
```



What is RabbitMQ

RabbitMQ is a complete, conformant and interoperable implementation of the published AMQP specification. It is licensed under the open source Mozilla Public License and has a platform-neutral distribution, plus platform-specific packages and bundles for easy installation.

Why choose RabbitMQ

RabbitMQ is based on a proven platform, offering exceptionally high reliability, availability and scalability along with good throughput and latency performance that is predictable and consistent. It has a compact, easily maintainable code base allowing rapid customisation and hot deployment. There are extensive facilities for management, monitoring, control and debugging and it is supported by a full range of commercial support services and an active community developing packages that extend the core system.

Read more about RabbitMQ and the company behind it.

Our news

- Catch up online
- Kaazing adds support for AMQP and RabbitMQ
- RabbitMQ 1.6.0 release

Distribution

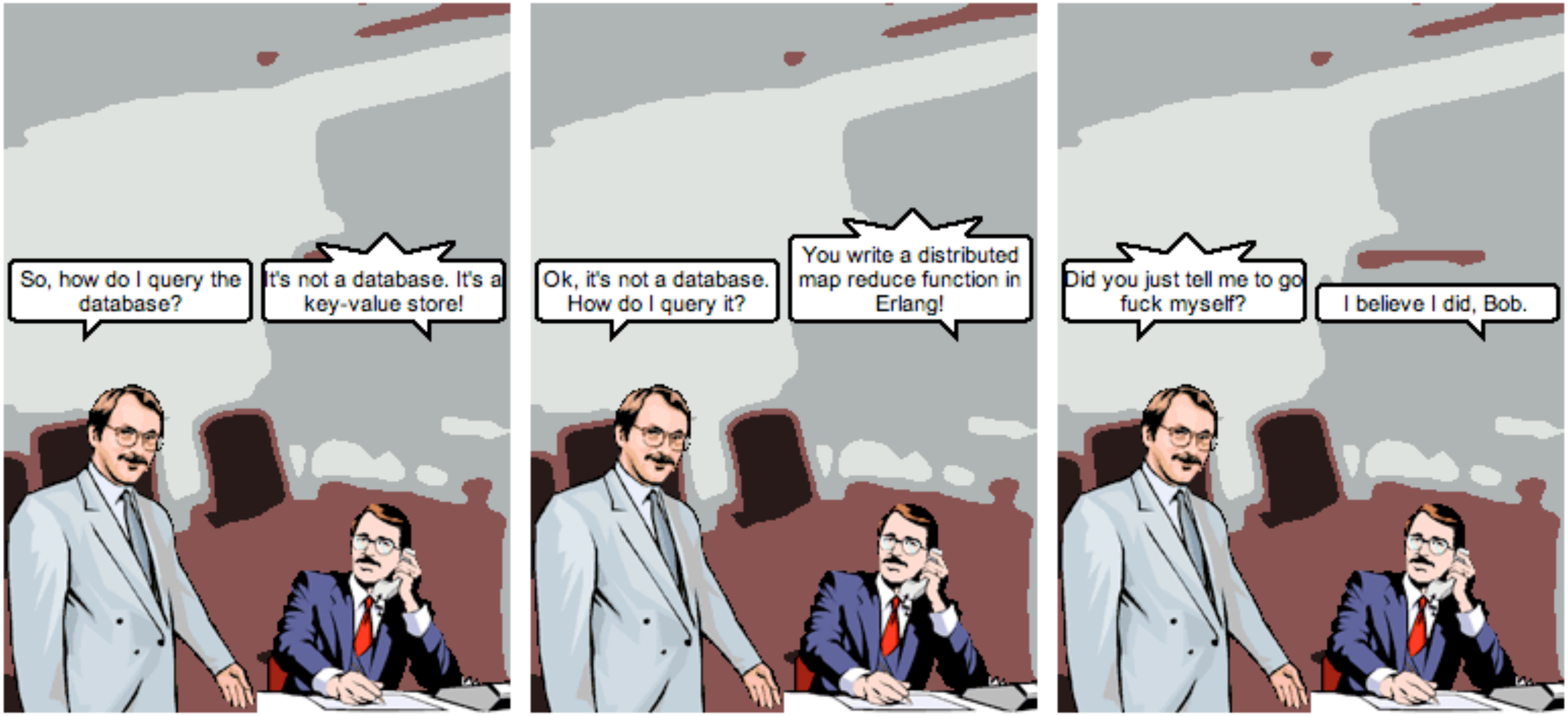
- RabbitMQ Java client with API guide, a .NET/C# client, with support for WCF, and many other clients available
- Experimental bindings supporting HTTP, STOMP, SMTP, POP3

How do I scale RabbitMQ?

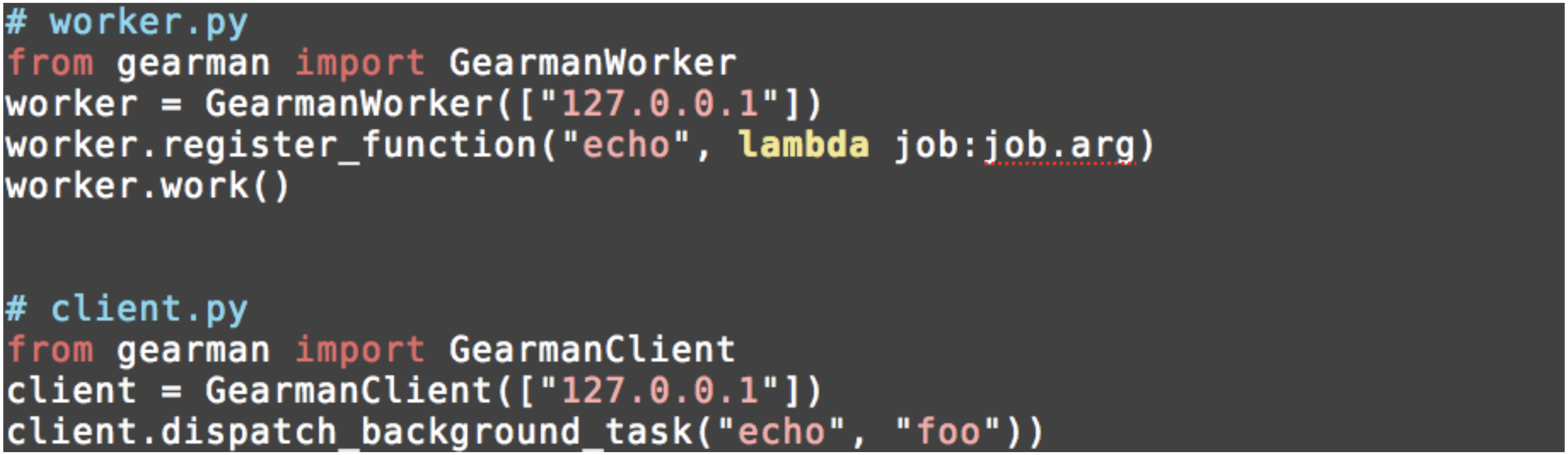
How do I scale RabbitMQ?

Use the power of Erlang

Fault-tolerance

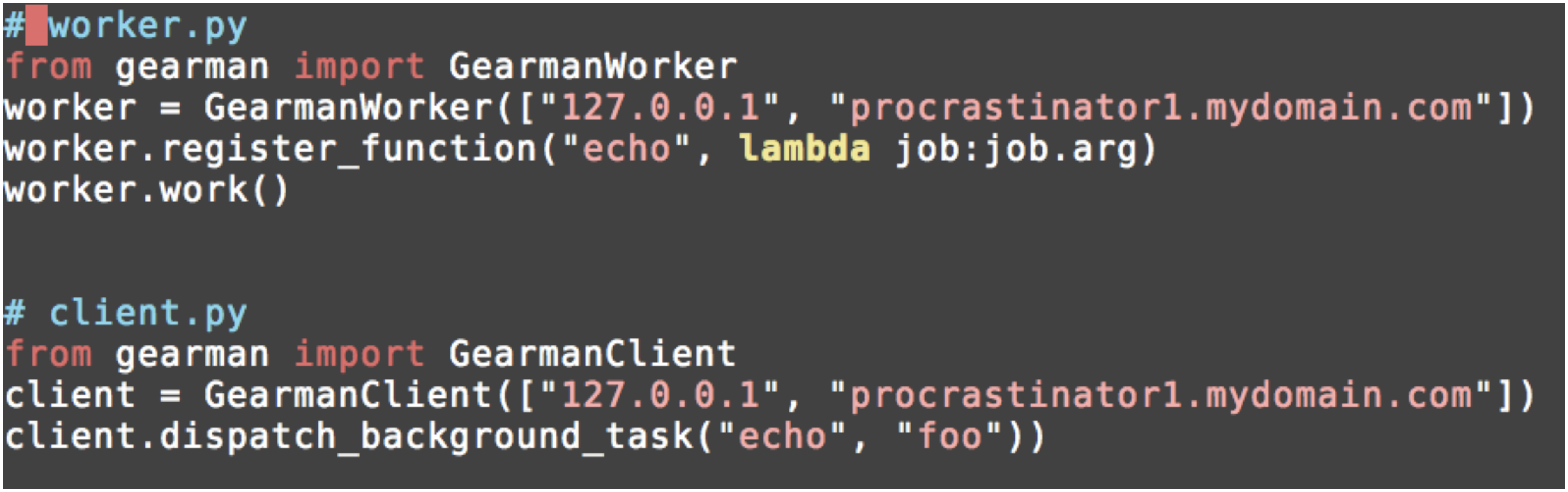

So, how do I query the database?
It's not a database. It's a key-value store!
Ok, it's not a database. How do I query it?
You write a distributed map reduce function in Erlang!
Did you just tell me to go fuck myself?
I believe I did, Bob.

How do I scale Gearman?



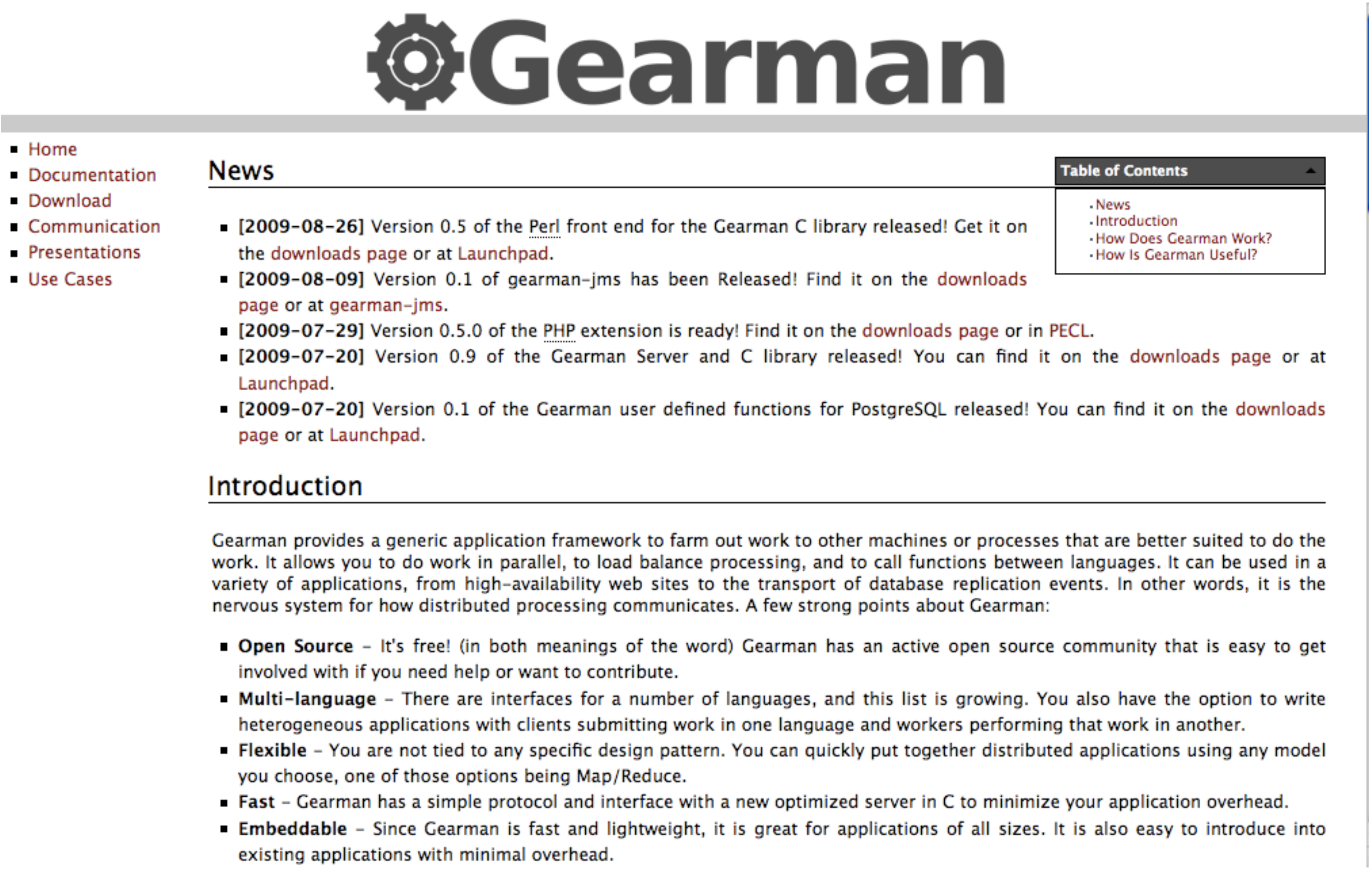
```python
# worker.py
from gearman import GearmanWorker
worker = GearmanWorker(["127.0.0.1"])
worker.register_function("echo", lambda job:job.arg)
worker.work()


# client.py
from gearman import GearmanClient
client = GearmanClient(["127.0.0.1"])
client.dispatch_background_task("echo", "foo"))
```



```
# worker.py
from gearman import GearmanWorker
worker = GearmanWorker(["127.0.0.1", "procrastinator1.mydomain.com"])
worker.register_function("echo", lambda job:job.arg)
worker.work()


# client.py
from gearman import GearmanClient
client = GearmanClient(["127.0.0.1", "procrastinator1.mydomain.com"])
client.dispatch_background_task("echo", "foo"))
```



Gearman

- Home
- Documentation
- Download
- Communication
- Presentations
- Use Cases

News

Table of Contents

- **[2009-08-26]** Version 0.5 of the Perl front end for the Gearman C library released! Get it on the downloads page or at Launchpad.
- **[2009-08-09]** Version 0.1 of gearman-jms has been Released! Find it on the downloads page or at gearman-jms.
- **[2009-07-29]** Version 0.5.0 of the PHP extension is ready! Find it on the downloads page or in PECL.
- **[2009-07-20]** Version 0.9 of the Gearman Server and C library released! You can find it on the downloads page or at Launchpad.
- **[2009-07-20]** Version 0.1 of the Gearman user defined functions for PostgreSQL released! You can find it on the downloads page or at Launchpad.

Introduction

Gearman provides a generic application framework to farm out work to other machines or processes that are better suited to do the work. It allows you to do work in parallel, to load balance processing, and to call functions between languages. It can be used in a variety of applications, from high-availability web sites to the transport of database replication events. In other words, it is the nervous system for how distributed processing communicates. A few strong points about Gearman:

- **Open Source** – It's free! (in both meanings of the word) Gearman has an active open source community that is easy to get involved with if you need help or want to contribute.
- **Multi-language** – There are interfaces for a number of languages, and this list is growing. You also have the option to write heterogeneous applications with clients submitting work in one language and workers performing that work in another.
- **Flexible** – You are not tied to any specific design pattern. You can quickly put together distributed applications using any model you choose, one of those options being Map/Reduce.
- **Fast** – Gearman has a simple protocol and interface with a new optimized server in C to minimize your application overhead.
- **Embeddable** – Since Gearman is fast and lightweight, it is great for applications of all sizes. It is also easy to introduce into existing applications with minimal overhead.