

a

b



How is Prezi.com like FibPro?
How is Prezi like FibInc?



Prezi

Similar problems led us to split up monolithic web app.
2 generations of HTTP-based RPC systems used.
Log sink one of the first services split off of monolith.
RPC metadata (request id, source, etc) important debug tool.
Per-service monitoring



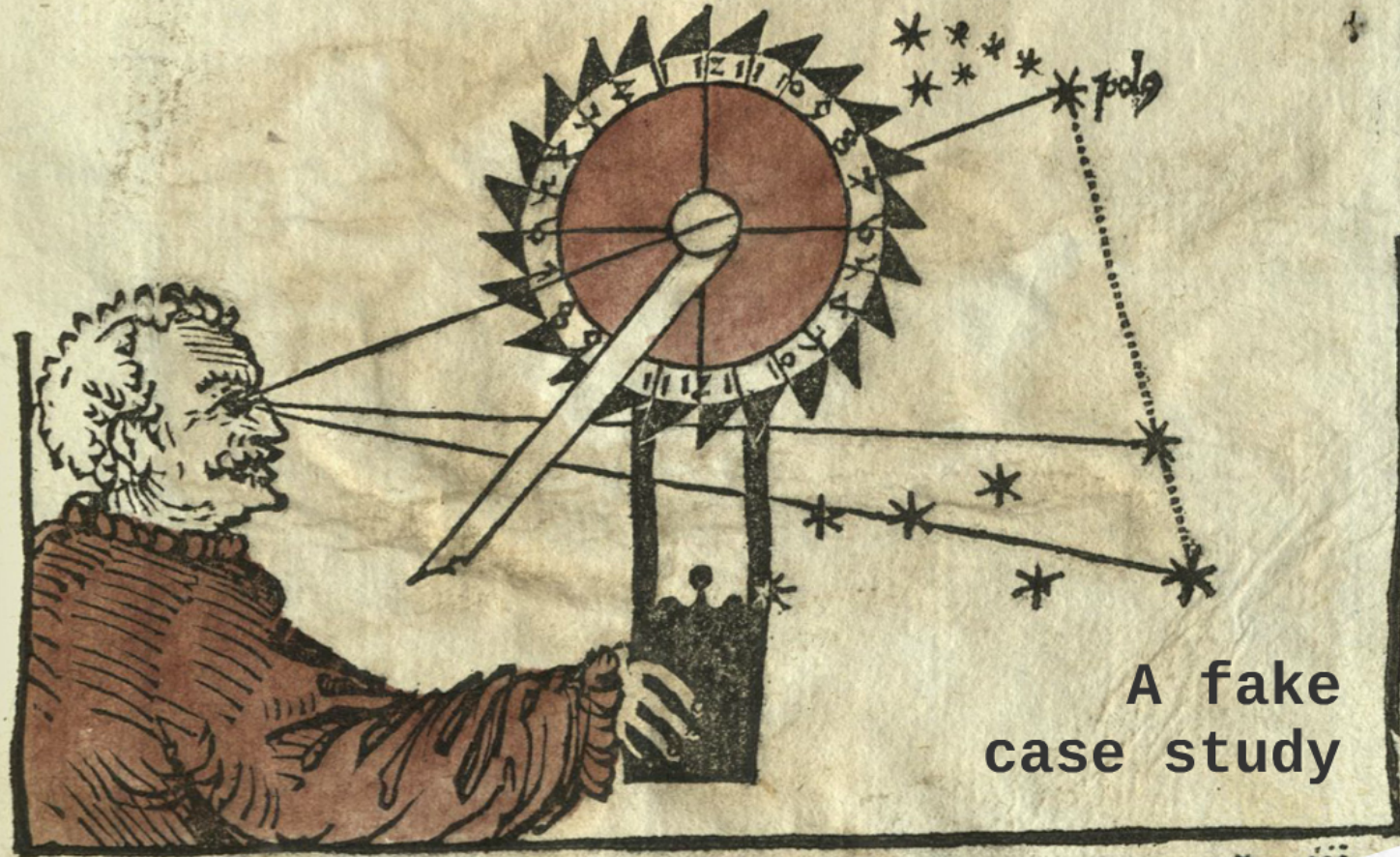
Thank you

Next steps for FibPro
API descriptions, versioning
service discovery
multiple environments
microservice tests

Vt Evidentissime Patet
In Figura Sequenti.

Ecce figuram.

Debugging Microservices



A fake
case study

Fibonacci



Entrepreneur profile

Leonardo Bonacci

Mathematician,
Engineer,
Co-founder and CTO
of FibInc.

Company profile

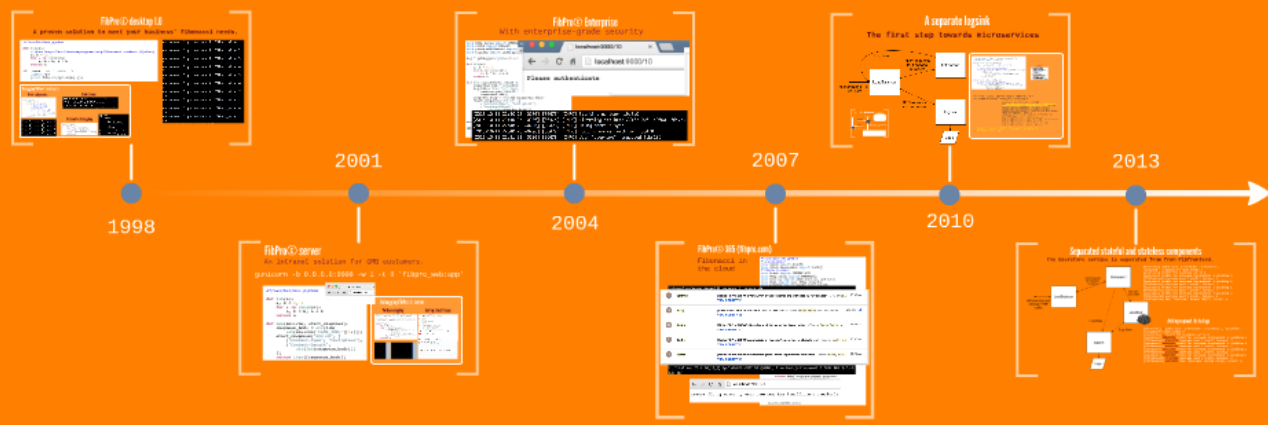
FibInc Corporation

Creator of FibPro®.
Today, the worldwide
leader in the
Fibonacci as a Service
(FaaS) market.

The evolution of FibPro®



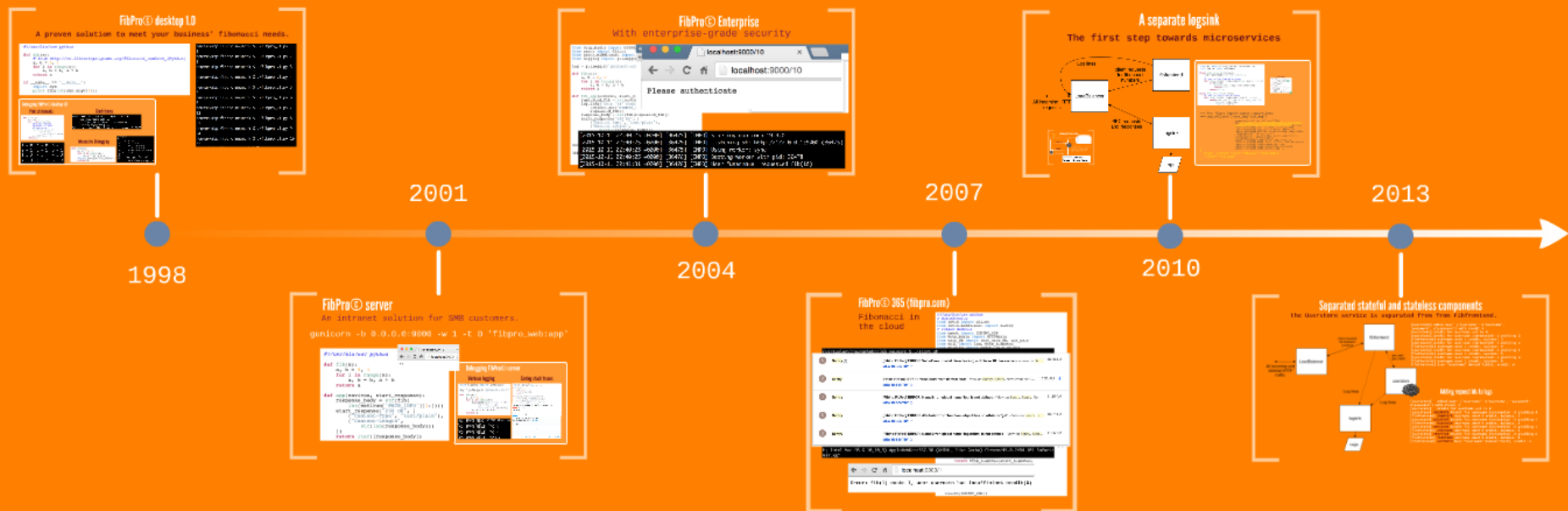
The evolution of FibPro©



Summary of problems and solutions

- high code complexity
- splitting the app
- Logs everywhere
- unified logging
- failed requests
- RemoteExceptions
- Grouping events
- request ids

The evolution of FibPro©



FibPro[©] desktop 1.0

A proven solution to meet your business' fibonacci needs.

```
#!/usr/bin/env python

def fib(n):
    # from http://en.literateprograms.org/Fibonacci_numbers_(Python)
    a, b = 0, 1
    for i in range(n):
        a, b = b, a + b
    return a

if __name__ == "__main__":
    import sys
    print fib(int(sys.argv[1]))
```

Debugging FibPro[©] desktop 1.0

Print statements

```
def fib(n):
    a, b = 0, 1
    for i in range(n):
        # print internal
        # state for each
        # iteration
        print ("i = %s, " +
              "a = %s, " +
              "b = %s") % (i,a,b)
        a, b = b, a + b
    return a
```

```
i = 0, a = 0, b = 1
i = 1, a = 1, b = 1
i = 2, a = 1, b = 2
i = 3, a = 2, b = 3
```

Stack traces

```
neumarkmbp:fibpro neumark $ ./fibpro_v1.py 4.5
Traceback (most recent call last):
  File "./fibpro_v1.py", line 11, in <module>
    print fib(int(sys.argv[1]))
ValueError: invalid literal for int() with base 10: '4.5'
```

Interactive Debugging

```
def fib(n):
    a, b = 0, 1
    for i in range(n):
        a, b = b, a + b
    import pdb;pdb.set_trace()
    return a
```

```
-> return a
(Pdb) l
3     def fib(n):
4         a, b = 0, 1
5         for i in range(n):
6             a, b = b, a + b
7             import pdb;pdb.set_trace()
8         return a
9
10    if __name__ == "__main__":
11        import sys
12        print fib(int(sys.argv[1]))
```

```
neumarkmbp:fibpro neumark $ ./fibpro_v1.py 1
1
neumarkmbp:fibpro neumark $ ./fibpro_v1.py 2
1
neumarkmbp:fibpro neumark $ ./fibpro_v1.py 3
2
neumarkmbp:fibpro neumark $ ./fibpro_v1.py 4
3
neumarkmbp:fibpro neumark $ ./fibpro_v1.py 5
5
neumarkmbp:fibpro neumark $ ./fibpro_v1.py 6
8
neumarkmbp:fibpro neumark $ ./fibpro_v1.py 7
13
neumarkmbp:fibpro neumark $ ./fibpro_v1.py 8
21
neumarkmbp:fibpro neumark $ ./fibpro_v1.py 9
34
neumarkmbp:fibpro neumark $ ./fibpro_v1.py 10
55
```



```
if __name__ == "__main__":
    import sys
    print fib(int(sys.argv[1]))
```

Debugging FibPro[©] desktop 1.0

Print statements

```
def fib(n):
    a, b = 0, 1
    for i in range(n):
        # print internal
        # state for each
        # iteration
        print ("i = %s, " + \
              "a = %s, " + \
              "b = %s") % (i,a,b)
        a, b = b, a + b
    return a
```

```
i = 0, a = 0, b = 1
i = 1, a = 1, b = 1
i = 2, a = 1, b = 2
i = 3, a = 2, b = 3
```

Stack traces

```
neumarkmbp:fibpro neumark $ ./fibpro_v1.py 4.5
Traceback (most recent call last):
  File "./fibpro_v1.py", line 11, in <module>
    print fib(int(sys.argv[1]))
ValueError: invalid literal for int() with base 10: '4.5'
```

Interactive Debugging

```
def fib(n):
    a, b = 0, 1
    for i in range(n):
        a, b = b, a + b
    import pdb;pdb.set_trace()
    return a
```

```
-> return a
(Pdb) l
3     def fib(n):
4         a, b = 0, 1
5         for i in range(n):
6             a, b = b, a + b
7             import pdb;pdb.set_trace()
8 ->         return a
9
10    if __name__ == "__main__":
11        import sys
12        print fib(int(sys.argv[1]))
```

FibPro[©] server

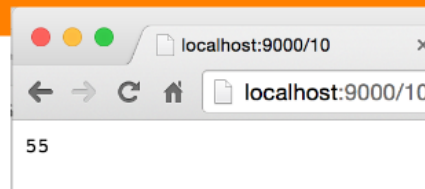
An intranet solution for SMB customers.

```
gunicorn -b 0.0.0.0:9000 -w 1 -t 0 'fibpro_web:app'
```

```
#!/usr/bin/env python
```

```
def fib(n):  
    a, b = 0, 1  
    for i in range(n):  
        a, b = b, a + b  
    return a
```

```
def app(environ, start_response):  
    response_body = str(fib(  
        int(environ['PATH_INFO'][1:])))  
    start_response("200 OK", [  
        ("Content-Type", "text/plain"),  
        ("Content-Length",  
         str(len(response_body)))  
    ])  
    return iter([response_body])
```



Debugging FibPro[©] server

Verbose logging

```
from logging import getLogger  
  
log = getLogger('gunicorn.error')  
  
def fib(n):  
    a, b = 0, 1  
    for i in range(n):  
        log.warn("i = %s, " + \  
            "a = %s, " + \  
            "b = %s" % (i,a,b))  
        a, b = b, a + b  
    return a
```

```
[2015-10-11 20:58:38 -0200] [5591] [INFO] Starting gunicorn 19.3.2  
[2015-10-11 20:58:38 -0200] [2892] [INFO] Listening at: http://0.0.0.0:9000 (1880)  
[2015-10-11 20:58:38 -0200] [2892] [INFO] Using worker: sync  
[2015-10-11 20:58:38 -0200] [2894] [INFO] Booting worker with pid: 1524  
[2015-10-11 20:58:52 -0200] [2894] [WARNING] - 0, a = 0, b = 1  
[2015-10-11 20:58:52 -0200] [2894] [WARNING] - 1, a = 1, b = 1  
[2015-10-11 20:58:52 -0200] [2894] [WARNING] - 2, a = 1, b = 2  
[2015-10-11 20:58:52 -0200] [2894] [WARNING] - 3, a = 2, b = 3  
[2015-10-11 20:58:52 -0200] [2894] [WARNING] - 4, a = 3, b = 5  
[2015-10-11 20:58:52 -0200] [2894] [WARNING] - 5, a = 5, b = 8  
[2015-10-11 20:58:52 -0200] [2894] [WARNING] - 6, a = 8, b = 13  
[2015-10-11 20:58:52 -0200] [2894] [WARNING] - 7, a = 13, b = 21  
[2015-10-11 20:58:52 -0200] [2894] [WARNING] - 8, a = 21, b = 34  
[2015-10-11 20:58:52 -0200] [2894] [WARNING] - 9, a = 34, b = 55
```

Saving stack traces

```
from raven import Client  
from raven.middleware import Sentry  
  
def fib(n):  
    a, b = 0, 1  
    for i in range(n):  
        a, b = b, a + b  
    return a  
  
def fib_app(environ, start_response):  
    response_body = str(fib(  
        int(environ['PATH_INFO'][1:])))  
    start_response("200 OK", [  
        ("Content-Type", "text/plain"),  
        ("Content-Length",  
         str(len(response_body)))  
    ])  
    return iter([response_body])
```

ValueError: Invalid literal for int() with base 10: '4.5'

fibpro_web.raven in fib_app

Details Comments Tips Show Events

Event 67e8031167041ed83f8a3799daadfd

October 11, 2015 7:25 PM UTC | [JSON](#) | [HTML](#)

Message

ValueError: Invalid literal for int() with base 10: '4.5'

Tags

Browser=Chrome-45.0 devtool=Other level=error os=Mac OS X 10.10.5

Debugging FibPro[©] server

Verbose logging

```
from logging import getLogger

log = getLogger('gunicorn.error')

def fib(n):
    a, b = 0, 1
    for i in range(n):
        log.warn(("i = %s, " + \
                "a = %s, " + \
                "b = %s") % (i, a, b))
        a, b = b, a + b
    return a
```

```
[2015-10-11 20:50:38 +0200] [35691] [INFO] Starting gunicorn 19.3.0
[2015-10-11 20:50:38 +0200] [35691] [INFO] Listening at: http://0.0.0.0:9000 (35691)
[2015-10-11 20:50:38 +0200] [35691] [INFO] Using worker: sync
[2015-10-11 20:50:38 +0200] [35694] [INFO] Booting worker with pid: 35694
[2015-10-11 20:50:52 +0200] [35694] [WARNING] i = 0, a = 0, b = 1
[2015-10-11 20:50:52 +0200] [35694] [WARNING] i = 1, a = 1, b = 1
[2015-10-11 20:50:52 +0200] [35694] [WARNING] i = 2, a = 1, b = 2
[2015-10-11 20:50:52 +0200] [35694] [WARNING] i = 3, a = 2, b = 3
[2015-10-11 20:50:52 +0200] [35694] [WARNING] i = 4, a = 3, b = 5
[2015-10-11 20:50:52 +0200] [35694] [WARNING] i = 5, a = 5, b = 8
[2015-10-11 20:50:52 +0200] [35694] [WARNING] i = 6, a = 8, b = 13
[2015-10-11 20:50:52 +0200] [35694] [WARNING] i = 7, a = 13, b = 21
[2015-10-11 20:50:52 +0200] [35694] [WARNING] i = 8, a = 21, b = 34
[2015-10-11 20:50:52 +0200] [35694] [WARNING] i = 9, a = 34, b = 55
```

Saving stack traces

```
from raven import Client
from raven.middleware import Sentry

def fib(n):
    a, b = 0, 1
    for i in range(n):
        a, b = b, a + b
    return a

def fib_app(environ, start_response):
    response_body = str(fib(
        int(environ['PATH_INFO'][:1])))
    start_response("200 OK", [
        ("Content-Type", "text/plain"),
        ("Content-Length",
```

ValueError: invalid literal for int() with base 10: '4.5'

Error fibpro_web_raven in fib_app



Details Comments 0 Tags Similar Events

Event 67e8d311da7041dd83f8d3799daad1fd

October 11, 2015 7:25 PM UTC | JSON (6.8 KB)

Message

ValueError: invalid literal for int() with base 10: 'favicon.ico'

Tags

browser = Chrome 45.0 device = Other level = error os = Mac OS X 10.10.5 s

FibPro[©] Enterprise

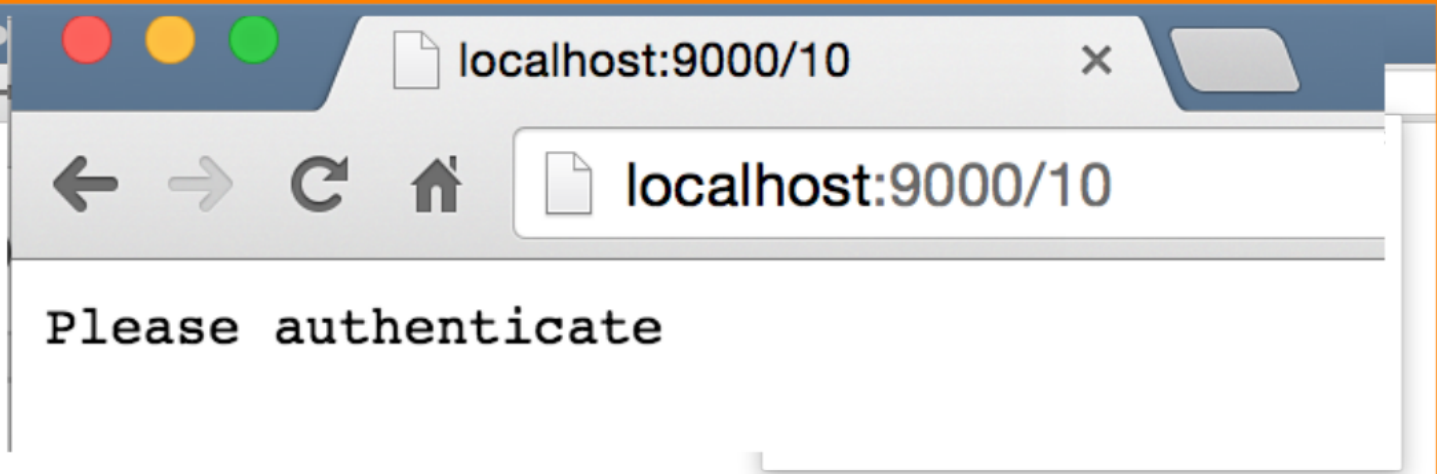
With enterprise-grade security

```
from http_basic import HTTPBasicAuth
from raven import Client
from raven.middleware import Raven
from logging import getLogger

log = getLogger('gunicorn.error')

def fib(n):
    a, b = 0, 1
    for i in range(n):
        a, b = b, a + b
    return a

def fib_app(environ, start_response):
    requested_fib = int(environ.get('REQUEST_URI', '').split('/')[-1])
    log.info("User \"%s\" requested fib(%d)", environ.get('REMOTE_ADDR', ' '),
            requested_fib)
    response_body = str(fib(requested_fib))
    start_response("200 OK", [
        ("Content-Type", "text/plain"),
        ("Content-Length", str(len(response_body)))
    ])
    return response_body
```














```
[2015-10-11 22:40:25 +0200] [36475] [INFO] Starting gunicorn 19.3.0
[2015-10-11 22:40:25 +0200] [36475] [INFO] Listening at: http://127.0.0.1:9000 (36475)
[2015-10-11 22:40:25 +0200] [36475] [INFO] Using worker: sync
[2015-10-11 22:40:25 +0200] [36478] [INFO] Booting worker with pid: 36478
[2015-10-11 22:41:31 +0200] [36478] [INFO] User "username" requested fib(10)
```


FibPro© 365 (fibpro.com)

Fibonacci in
the cloud

```
#!/usr/bin/env python
# dependencies
from raven import Client
from raven.middleware import Sentry
# fibpro modules
from const import SENTRY_DSN
from http_basic import HTTPBasic
from user_db import init_user_db, get_user
from util import log, http_response
from pricing import update_user_credit
```

```
(virtualenv)neumarkmbp:365 neumark $ ./start.sh
```

-  **Sentry** (4) [FibInc FibPro] ERROR: ValueError: invalid literal for int() with base 10: 'favicon.ico' – View on [Sentry](#) 10:08 AM
[VIEW IN SENTRY](#) 
-  **Sentry** [FibInc FibPro] ERROR: ValueError: math domain error – View on [Sentry](#). [Sentry](#). New Event on Fi... 10:33 AM 
[VIEW IN SENTRY](#) 
-  **Sentry** [FibInc FibPro] ERROR: NameError: global name 'log' is not defined – View on [Sentry](#). [Sentry](#). Ne... 11:25 AM
[VIEW IN SENTRY](#) 
-  **Sentry** [FibInc FibPro] ERROR: AttributeError: 'function' object has no attribute 'get' – View on [Sentry](#). [S...](#) 10:07 AM
[VIEW IN SENTRY](#) 
-  **Sentry** [FibInc FibPro] ERROR: NameError: global name 'logarithm' is not defined – View on [Sentry](#). [Sent...](#) 11:25 AM
[VIEW IN SENTRY](#) 

```
h; Intel Mac OS X 10_10_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/45.0.2454.101 Safari/537.36"
```

```
return http_response(start_response,
```

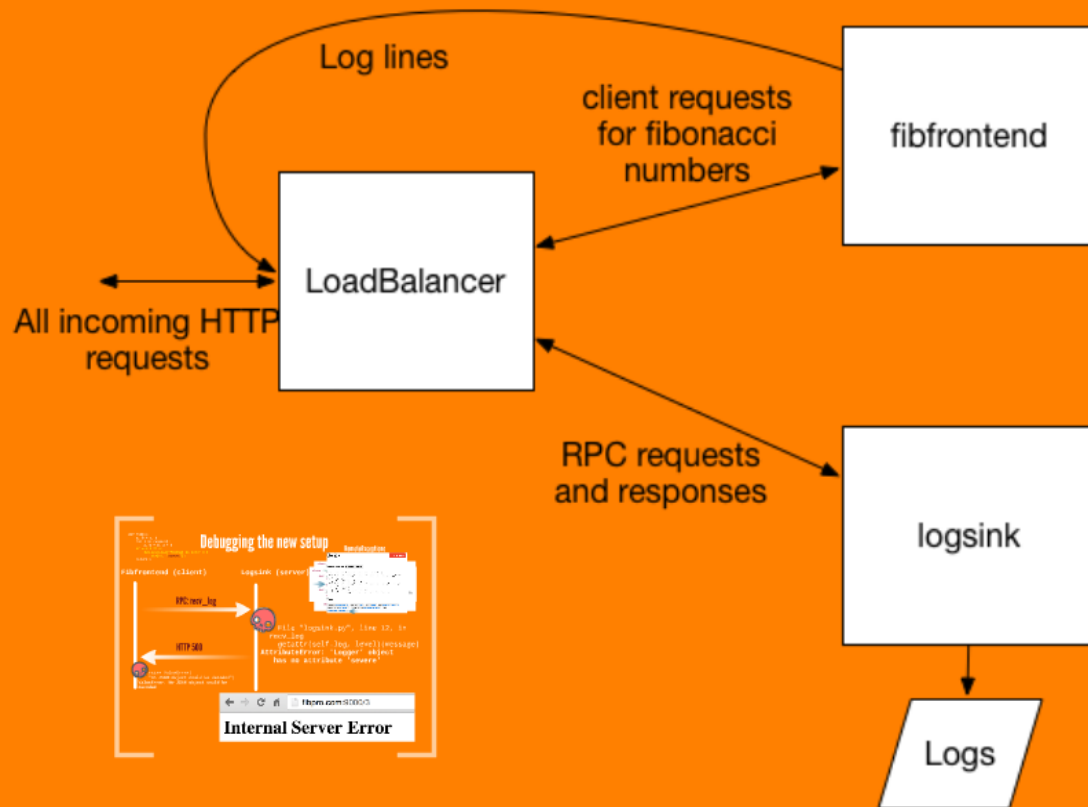
← → ↻ 🏠 📄 localhost:9000/1

Error: fib(1) costs 1, user username has insufficient credit(0)

```
Client(SENTRY_DSN)
```

A separate logsink

The first step towards microservices



```
from rpc import Client, Server, ServerConfig
from logging import getLogger

class LogSinkServer(Server):
    log = getLogger('unicorn.error')

    # runs in the logsink process
    def recv_log(self, message=None, level='info'):
        if message:
            getattr(self.log, level)(message)
            return True
        return False

class LogSinkClient(Client):
    # runs in the client process
    def send_log(self, message, level='info'):
        return self.call('logsink', 'recv_log', {
            'message': message,
            'level': level})
```



```
>>> from fibpro.logsink import LogSinkClient
>>> LogSinkClient().send_log("test_msg")

accept[0004]=0005 from [127.0.0.1:53521]
GET /api/logsink/?req=eyJhcwdrIjpw7lswldmV1joiawWbhyisim1o3 Wlz2uio13ozxw0x2iszzy39tC3tZXhobzQ101jyZmxxzsvzy39 HTTP/1.1
http-in.clihdr[0005:ffffff]: Host: fibpro.com:9000
http-in.clihdr[0005:ffffff]: Connection: keep-alive
http-in.clihdr[0005:ffffff]: Accept-Encoding: gzip, deflate
clihdr[0005:ffffff]: Accept: */*
clihdr[0005:ffffff]: User-Agent: python-requests/2.8.0
srvrep[0005:0006]: HTTP/1.1 200 OK
logsink.srvhdr[0005:0006]: Server: unicorn/19.3.0
logsink.srvhdr[0005:0006]: Date: Tue, 13 Oct 2015 00:34:04 GMT
logsink.srvhdr[0005:0006]: Connection: close
logsink.srvhdr[0005:0006]: Content-Type: text/plain
logsink.srvhdr[0005:0006]: Content-Length: 4

{"args": {"level": "info", "message": "test_msg"},
 "method": "recv_log"}
```



```

from rpc import Client, Server, ServerConfig
from logging import getLogger

class LogSinkServer(Server):
    log = getLogger('unicorn.error')

    # runs in the logsink process
    def recv_log(self, message=None, level='info'):
        if message:
            getattr(self.log, level)(message)
            return True
        return False

class LogSinkClient(Client):
    # runs in the client process
    def send_log(self, message, level='info'):
        return self.call('logsink', 'recv_log', {
            'message': message,
            'level': level})

```

The RPC "framework"

```

class Server(RPCBase):
    def __init__(self, environ, start_response):
        method, args = self.decode_arguments(
            environ['REQUEST_METHOD'], environ['REQUEST_URI'])
        return self.call(self, method, args)
    def decode_arguments(self, method, args):
        return self.decode_result(returned)

class Client(RPCBase):
    def __init__(self, server_config=None):
        self.server_config = server_config or ServerConfig()
    def call(self, service, method, args):
        return self.config.endpoint(service,
            self.server_config.endpoint(service),
            self.config.protocol(service, args))
    def __call__(self, service, method, args=None):
        url = self.config.url(service, method, args or ())
        return self.decode_result(request.urlopen(url).read())

```

```

>>> from fibpro.logsink import LogSinkClient
>>> LogSinkClient().send_log("test_msg")

```

```

accept(0004)=0005 from [127.0.0.1:53521]
GET /api/logsink/?req=eyJhc2VudDp7ImxldmVsIjoiaW5mbyIsIm1lc3
NhZ2UiOiJ0ZXN0X2IzZyJ9LCJtZXRob2QiOiJyZWZWN2X2xvZyJ9 HTTP/1.1
http-in.clihdr[0005:ffffffff]: Host: fibpro.com:9000
http-in.clihdr[0005:ffffffff]: Connection: keep-alive
http-in.clihdr[0005:ffffffff]: Accept-Encoding: gzip, deflate
clihdr[0005:ffffffff]: Accept: */*
clihdr[0005:ffffffff]: User-Agent: python-requests/2.8.0
srvrep[0005:0006]: HTTP/1.1 200 OK
logsink.srvhdr[0005:0006]: Server: unicorn/19.3.0
logsink.srvhdr[0005:0006]: Date: Tue, 13 Oct 2015 09:34:04 GMT
logsink.srvhdr[0005:0006]: Connection: close
logsink.srvhdr[0005:0006]: Content-Type: text/plain
logsink.srvhdr[0005:0006]: Content-Length: 4

```

```

{
  "args": {"level": "info", "message": "test_msg"},
  "method": "recv_log"
}

```

The RPC "framework"

```
class Server(RPCBase):
```

```
    def wsgi_app(self, environ, start_response):
        method, args = self.decode_arguments(
            environ['QUERY_STRING'][(len(self.ARG_PREFIX)-1):])
        returned = getattr(self, method)(*args)
        return http_response(start_response,
            body=self.encode_result(returned))
```

```
    def app(self):
        return Sentry(self.wsgi_app,
            RavenClient(SENTRY_DSN))
```

```
class Client(RPCBase):
```

```
    def __init__(self, server_config=None):
        self.server_config = server_config or ServerConfig()
```

```
    def construct_url(self, service, method, args):
        return "%s%s%s" % (
            self.server_config.endpoints[service],
            self.ARG_PREFIX,
            self.encode_arguments(method, args))
```

```
    def call(self, service, method, args=None):
        url = self.construct_url(service, method, args or {})
        return self.decode_result(requests.get(url).text)
```

```
def fib(n):
    a, b = 0, 1
    for i in range(n):
        a, b = b, a + b
        if a % 2 == 0:
            log.send_log("fib(%s) is even" % (
                str(n), "severe"))
    return a
```

Debug

Fibfrontend (client)

Debugging the new setup

```
def fib(n):  
    a, b = 0, 1  
    for i in range(n):  
        a, b = b, a + b  
    if a % 2 == 0:  
        log.send_log("fib(%s) is even" % (  
            str(n), "severe"))  
    return a
```

Fibfrontend (client)

Logsink (server)

RPC: recv_log

HTTP 500



```
raise ValueError(  
    "No JSON object could be decoded")  
ValueError: No JSON object could be  
decoded
```



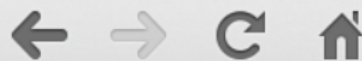
File "logsink.py", line 12, in
recv_log
 getattr(self.log, level)(message)
AttributeError: 'Logger' object
has no attribute 'severe'

RemoteExceptions

Sentry

View on Sentry

```
class  
New Event on FibInc FibPro  
class Cli  
RemoteException: Traceback (most recent call last):  
  File "/Users/neumark/git/fibpro-microservices/fibpro/rpc.py", line  
  175, in wsgi_app  
    getattr(self, method)(*args)  
  File "/Users/neumark/git/fibpro-microservices/fibpro/logsink.py",  
  line 12, in recv_log  
  AttributeError: 'Logger' object has no attribute 'severe'  {}  
  {}  
Tags  
de  
browser = Chrome 45.0 device = Other level = error os = Mac OS X 10.10.5  
sentry:user = ip:127.0.0.1 server_name = neumarkmbp.local url =  
http://fibpro.com:9000/3
```



fibpro.com:9000/3

Internal Server Error

RemoteExceptions

Sentry

[View on Sentry](#)

class

New Event on FibInc FibPro

class Cli

RemoteException: Traceback (most recent call last):

```
def r File "/Users/neumark/git/fibpro-microservices/fibpro/rpc.py", line
i75, in wsgi_app
```

```
    getattr(self, method)(**args))
```



```
r File "/Users/neumark/git/fibpro-microservices/fibpro/logsink.py",
line 12, in recv_log
```

```
def c    getattr(self.log, level)(message)
```

```
AttributeError: 'Logger' object has no attribute 'severe'
```

```
{})
```

Tags

de

browser = **Chrome 45.0**

device = **Other**

level = **error**

os = **Mac OS X 10.10.5**

sentry:user = **ip:127.0.0.1**

server_name = **neumarkmbp.local**

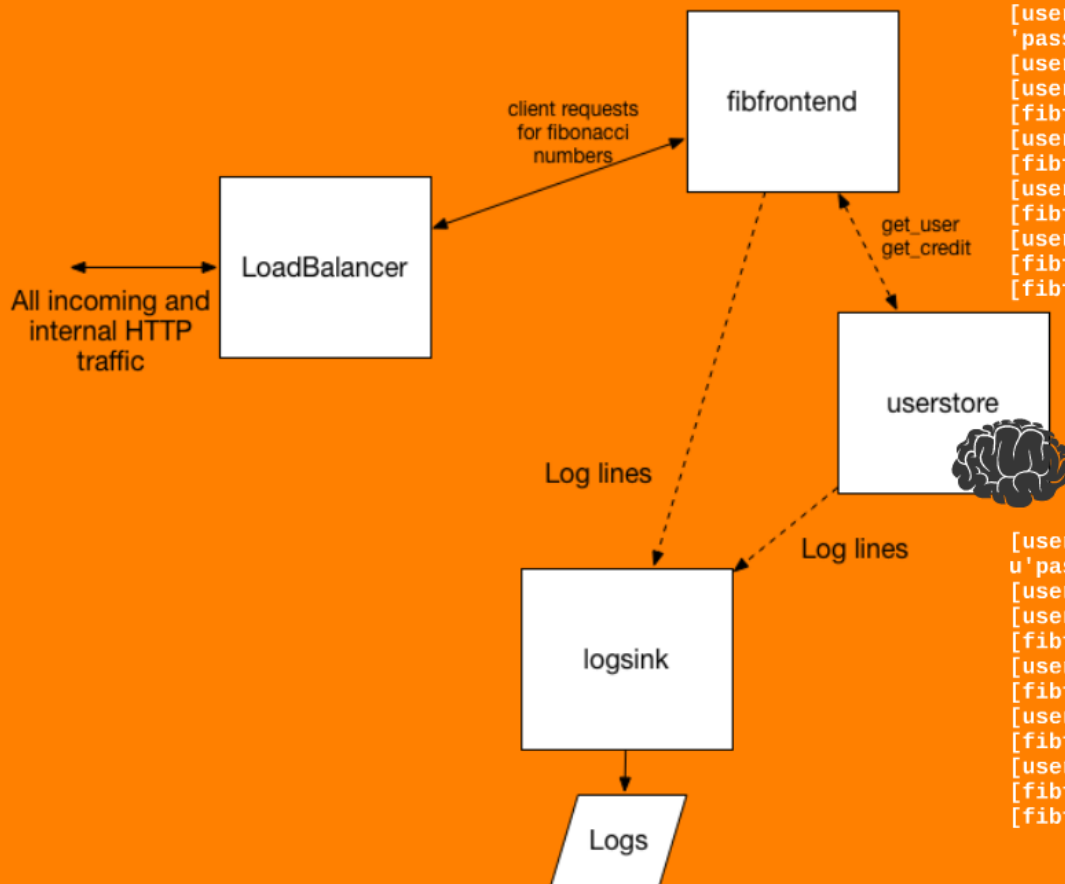
url =

http://fibpro.com:9000/3



Separated stateful and stateless components

the Userstore service is separated from from Fibfrontend.



```
[userstore] Added user: {'username': u'username', 'password': u'password'} with credit 4
[userstore] Credit for username set to 4
[userstore] Credit for username incremented -1 yielding 3
[fibfrontend] username used 1 credit, balance: 3
[userstore] Credit for username incremented -1 yielding 2
[fibfrontend] username used 1 credit, balance: 2
[userstore] Credit for username incremented -1 yielding 1
[fibfrontend] username used 1 credit, balance: 1
[userstore] Credit for username incremented -1 yielding 0
[fibfrontend] username used 1 credit, balance: 0
[fibfrontend] User "username" denied fib(3), credit: 0
```

Adding request ids to logs

```
[userstore] - Added user: {'username': u'username', 'password': u'password'} with credit 4
[userstore] - Credit for username set to 4
[userstore] 3b9d41f8 Credit for username incremented -1 yielding 3
[fibfrontend] 3b9d41f8 username used 1 credit, balance: 3
[userstore] 0a5eb154 Credit for username incremented -1 yielding 2
[fibfrontend] 0a5eb154 username used 1 credit, balance: 2
[userstore] a9ee3a98 Credit for username incremented -1 yielding 1
[fibfrontend] a9ee3a98 username used 1 credit, balance: 1
[userstore] 7642f208 Credit for username incremented -1 yielding 0
[fibfrontend] 7642f208 username used 1 credit, balance: 0
[fibfrontend] aaff62f3 User "username" denied fib(3), credit: 0
```



```
[userstore] Added user: {'username': u'username',  
'password': u'password'} with credit 4  
[userstore] Credit for username set to 4  
[userstore] Credit for username incremented -1 yielding 3  
[fibfrontend] username used 1 credit, balance: 3  
[userstore] Credit for username incremented -1 yielding 2  
[fibfrontend] username used 1 credit, balance: 2  
[userstore] Credit for username incremented -1 yielding 1  
[fibfrontend] username used 1 credit, balance: 1  
[userstore] Credit for username incremented -1 yielding 0  
[fibfrontend] username used 1 credit, balance: 0  
[fibfrontend] User "username" denied fib(3), credit: 0
```

get_user
get_credit

userstore



Adding request ids to logs

```
lines [userstore] - Added user: {'username': u'username', 'password':  
u'password'} with credit 4  
[userstore] - Credit for username set to 4  
[userstore] 3b9d41f8 Credit for username incremented -1 yielding 3  
[fibfrontend] 3b9d41f8 username used 1 credit, balance: 3  
[userstore] 0a5eb154 Credit for username incremented -1 yielding 2  
[fibfrontend] 0a5eb154 username used 1 credit, balance: 2  
[userstore] a9ee3a98 Credit for username incremented -1 yielding 1  
[fibfrontend] a9ee3a98 username used 1 credit, balance: 1  
[userstore] 7642f208 Credit for username incremented -1 yielding 0  
[fibfrontend] 7642f208 username used 1 credit, balance: 0  
[fibfrontend] aaff62f3 User "username" denied fib(3), credit: 0
```

Summary of problems and solutions

high code complexity splitting the app

Logs everywhere unified logging

failed requests RemoteExceptions

Grouping events request ids

2 gener
Log sim
RPC me
Per-ser




Monitoring

```
from rpc import Client
client = Client()
services = client.server_config.get_services()
for service in services:
    print service, client.call('ping',
    {}, service)
```



watch -n 1 python fibpro/monitor.py
Typically monitored:
%exceptions **#requests** **call time**




Phonetic

Entrepreneur profile
Leonardo Bonacci

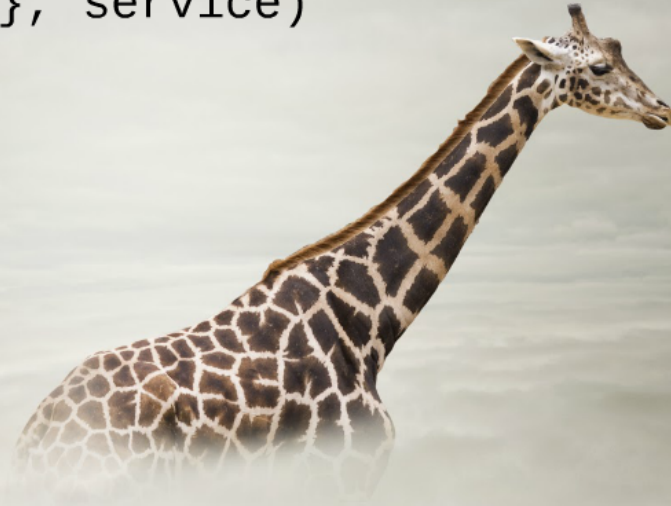
Mathematician,
Engineer,
Co-founder and CTO
of FibInc.

Company profile
FibInc Corporation
Creator of FibPro.
Today, the worldwide
leader in the
FibProsci of a Service
(FaaS) market.



Monitoring

```
from rpc import Client
client = Client()
services = client.server_config.get_services()
for service in services:
    print service, client.call('ping',
                               {}, service)
```



watch -n 1 python fibpro/monitor.py

Typically monitored:

#requests

%exceptions

call time

a



b

How is Prezi.com like FibPro?
How is Prezi like FibInc?



Prezi

Similar problems led us to split up monolithic web app.
2 generations of HTTP-based RPC systems used.
Log sink one of the first services split off of monolith.
RPC metadata (request id, source, etc) important debug tool.
Per-service monitoring



Thank you

Next steps for FibPro
API descriptions, versioning
service discovery
multiple environments
microservice tests

How is Prezi.com like FibPro? How is Prezi like FibInc?



Prezi

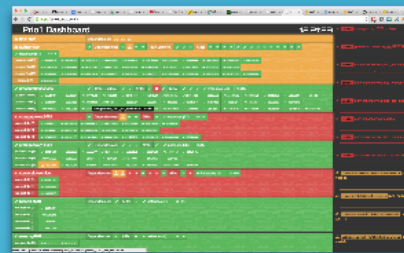
Similar problems led us to split up monolithic web app.

2 generations of HTTP-based RPC systems used.

Log sink one of the first services split off of monolith.

RPC metadata (request id, source, etc) important debug tool.

Per-service monitoring



Browser tabs: #270, Paym, local, oam, Smar, Get S, Codii, Mátý, prese, rende, Main, Prio1, AWS, resize, AWS, client, Dyna

Address bar: https://prio1.prezi.com/#

Prio1 Dashboard

12:21:39

prio1path Dependencies: ⚠️ ⚠️ ⚠️

authservice Dependencies: ✓ ✓ ✓ ✓ Rate-Limiters: ✓ ✓ ✓ ✓ APIs: ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓

Instances (31): 81.47%

us-east-1e (11)	✓ I-aaaxw97d	✓ I-7bc73c99	✓ I-f5858b1a	✓ I-eb858b04	✓ I-2fa433ce	✓ I-fb19650a	✓ I-2539cdo4	✓ I-e9858b06	✓ I-35dd64c4	✓ I-b123945f	✓ I-e8249309
us-east-1b (10)	✓ I-d0b67d3b	✓ I-08b44fa3	✓ I-1decf4e6	✓ I-6948118d	✓ I-018fa2fb	✓ I-79e15d83	✓ I-145c3a1f	✓ I-5fbcd78b4	✓ I-c659eb3c	✓ I-fad1ae10	
us-east-1c (11)	✓ I-85a06a69	✓ I-0eb43fa3	✓ I-99b7ba74	✓ I-5e6bdf73	✓ I-a202c64d	✓ I-9f6bdf72	✓ I-156ade38	✓ I-327897dd	✓ I-176ede3a	✓ I-b1a7c75c	✓ I-8659996a
preprod (1)	✓ I-a338370f										

presentationsservice Rate-Limiters: ✓ ✓ APIs: ✓ ✗ Misc: ✓ ✓ Instances (40): 76.66%

us-east-1b (14)	✓ I-7aa06191	✓ I-6ec38084	✓ I-89772b63	✓ I-729fb398	✓ I-851b460f	✓ I-1e914b34	✓ I-9c6258b1	✓ I-b89cdf52	✓ I-5399dab9	✓ I-a1500c4b	✓ I-88954fa2	✓ I-610451b5	✓ I-ad9ac947	✓ I-6fb5f094
us-east-1c (13)	✓ I-2c259301	✓ I-30a27edc	✓ I-68438884	✓ I-021ebbee	✓ I-cdcd0721	✓ I-2c8821c1	✓ I-03ae02ef	✓ I-6c00c541	✓ I-f58e70b8	✓ I-1dda7ef1	✓ I-58e59ef4	✓ I-ee906ec7	✓ I-96c71b7a	
us-east-1e (14)	✓ I-6114b080	✓ I-87b538ab	✓ I-ab47	storageproxy_0701_dep_dynamodb: 1m 10s	✓ I-4b6bd067	✓ I-73365092	✓ I-8ca2cc72	✓ I-7485c195	✓ I-b114b250	✓ I-4a8bd066	✓ I-7c74cd50	✓ I-32a0c6d3		

storageproxy_0701 Dependencies: ⚠️ ✓ ✓ APIs: ✓ ✓ Instances (21): 50.73%

us-east-1c (7)	✓ I-dc099a31	✓ I-26b71ad9	✓ I-9005967d	✓ I-6b375086	✓ I-66d67b99	✓ I-05bc24e8	✓ I-c236512f					
us-east-1b (6)	✓ I-8c306e61	✓ I-92336c78	✓ I-c4326c3e	✓ I-dfff8224	✓ I-e369721e	✓ I-98306e7c						
us-east-1e (8)	✓ I-65e23395	✓ I-aedacc05	✓ I-1ba233eb	✓ I-60a491b6	✓ I-cb22623b	✓ I-f82db116	✓ I-4e640099	✓ I-7738a2a0				

storageproxy_0901 Dependencies: ✓ ✓ APIs: ✓ ✓ Instances (23): 53.6%

us-west-2c (8)	✓ I-96021a99	✓ I-e88b92e9	✓ I-baaaa477	✓ I-bd041cb2	✓ I-ed8a93a2	✓ I-00021a0f	✓ I-ec8891a3	✓ I-358d943a				
us-west-2b (7)	✓ I-99107093	✓ I-28ba3922	✓ I-5e75f554	✓ I-26ff79d1	✓ I-4c744447	✓ I-a172f2ab	✓ I-56ef6f5c					
us-west-2a (8)	⚠️ I-f892d03	✓ I-4d973341	✓ I-148e2ef8	✓ I-8e943082	✓ I-4e973342	✓ I-dee146d2	✓ I-dbb12d2d	✓ I-55953159				

conversionsservice Dependencies: ⚠️ ⚠️ ✗ ✗ APIs: ✓ ✗ Instances (3): 51.53%

us-east-1b (1)	✓ I-9bd91161							
us-east-1e (1)	✓ I-br82894e							
us-east-1c (1)	✓ I-625f180							

rosetta_0701 Dependencies: ✓ APIs: ✓ Instances (4): 53.16%

us-east-1b (1)	✓ I-8b9e6e81						
us-east-1c (1)	✓ I-2619a1e						
us-east-1a (1)	✓ I-9baac9b4						
us-east-1e (1)	✓ I-bcd3125d						

rosetta_0901 Dependencies: ✓ APIs: ✓ Instances (4): 52.22%

us-west-2a (2)	✓ I-5797fb5b	✓ I-ecac7be0					
----------------	--------------	--------------	--	--	--	--	--

Alerts on the right:

- oam3.us storageproxy_0701_outage
- oam3.us presentationsservice_api_RESTAPI
- oam3.us conversionsservice_dep_worker_defa
- oam3.us conversionsservice_dep_worker_exp
- oam3.us conversionsservice_dep_sqs_size
- oam3.us conversionsservice_outage
- oam3.us conversionsservice_dep_worker_url
- oam3.us conversionsservice_api_api_export_w
- uptimetest-eu-west-1-f4d58210.ec2-eu-west-1a prio1|west-1a
- crunner01-i-9efbf60.ec2-us-east-1e prio1|path_out
- uptimetest-us-east-1-720c4aa5.ec2-us-east-1e prio1|east-1e
- uptimetest-us-west-1-79460db0.ec2-us-west-2c prio1|west-2c

Footer: https://prio1.prezi.com/#show/storageproxy_0701/storageproxy_0701_dep_dynamodb



A white Fibonacci spiral graphic is positioned on the left side of the slide, set against a blue background. The spiral starts from a small square and grows as it moves clockwise, with each new square's side length equal to the sum of the two preceding squares. The spiral is contained within a grid of white lines.

Thank you

Next steps for FibPro

API descriptions, versioning

service discovery

multiple environments

microservice tests