

2ndQuadrant

Professional PostgreSQL

www.2ndQuadrant.com



Barmman

Backup and recovery
manager for PostgreSQL

GABRIELE BARTOLINI

PostgreSQL Sessions - Paris, 4 October 2012

GABRIELE BARTOLINI

- Co-Founder and Manager of 2ndQuadrant Italia
 - Data Architect, Business critical environments
 - Data warehousing
- Co-Founder Italian PostgreSQL Users Group
- Co-Founder PostgreSQL Europe
- PostgreSQL Contributor and Advocate

INGRES

Database Now

French Painter 19th century



BARTOLINI

Myself

Italian Sculptor
(*same family tree*)

Now, here

19th century

1799

PARIS

BARTOLINI

INGRES

2012

PARIS

BARTOLINI

POSTGRES

DISCLAIMER

This talk assumes you are familiar with disaster recovery concepts and PostgreSQL implementation of Point In Time Recovery

BE AWARE

In 2ndQuadrant, all these concepts usually fit in a 2 day workshop on Disaster Recovery and a 1 day workshop on Barman alone

MENU D'AUJOURD'HUI

- Business continuity / Disaster recovery for databases
- Disaster recovery with Barman for PostgreSQL

PART I

Business continuity / Disaster recovery for databases

BUSINESS CONTINUITY

activity performed by an organization to ensure that **critical business** functions will be **available** to customers, suppliers, regulators, and other entities that must have access to those functions - Wikipedia

INFORMATION TECHNOLOGY

- Business continuity
 - High availability
 - **Disaster recovery**

LAW REQUIREMENTS

In Italy, the “**Codice dell’Amministrazione Digitale**” defines **business continuity requirements** for public administrations

DISASTER

(touch wood)

system/hardware failures
accidental errors
disaster

REACT TO A DISASTER

Recover systems, **data** and infrastructures

TOO LATE!

Do not wait for a disaster to happen

PLAN FOR DISASTERS

“Disasters” will happen. **Be prepared.**

*“Plans are worthless, but **planning is everything**. There is a very great distinction because when you are planning for an emergency you must start with this one thing: the very definition of **“emergency”** is that it is **unexpected**, therefore it is not going to happen the way you are planning.”*

- Dwight D. Eisenhower

REGULAR CRASH TESTS

DATABASE DISASTER RECOVERY

Let's just focus on **databases!**

REQUIREMENTS

- Automated backups
- Notifications (anomalies)
- Frequency of backups
- Retention policies
- Data protection
- Availability for recovery

POSTGRES BACKUP

- Hot backup
 - MVCC
- Logical Backup
 - pg_dump
- Physical Backup
 - Full backup (base backup)
 - Differential backup (WAL)

TRADITIONAL DR WITH POSTGRESQL

- PostgreSQL primitives for DR are robust and reliable
- High level skills
 - DBA
 - Sysadmins
- **Custom scripts**
- Hard to integrate in:
 - Backup solutions
 - Disaster Recovery plans
- **Hard to test!**

EXISTING TOOLS

- Omni-PITR
 - WAL centric
- WALmgr
 - WAL centric
- pg-rman
 - Server centric
- WAL-E
 - EC2 centric, but ...
 - good
 - came later

NONE FOR DR

None of them was a pure **disaster recovery** solution.
We wanted something similar to Oracle's RMAN.

FILLING A HOLE

The lack of a DR solution is a **barrier** towards the **adoption** of PostgreSQL from Oracle users' point of view.

DESIDERATA

- Hot, Full, Differential and incremental backups
- Multiple servers
- Remote backup & recovery
- Backup catalogues
- Retention policies
- Archival and compression
 - WAL segments
 - Periodical backups
- **Automation**
- **Integration**
- **Usability**

WWW.PGBARMAN.ORG



Barmman

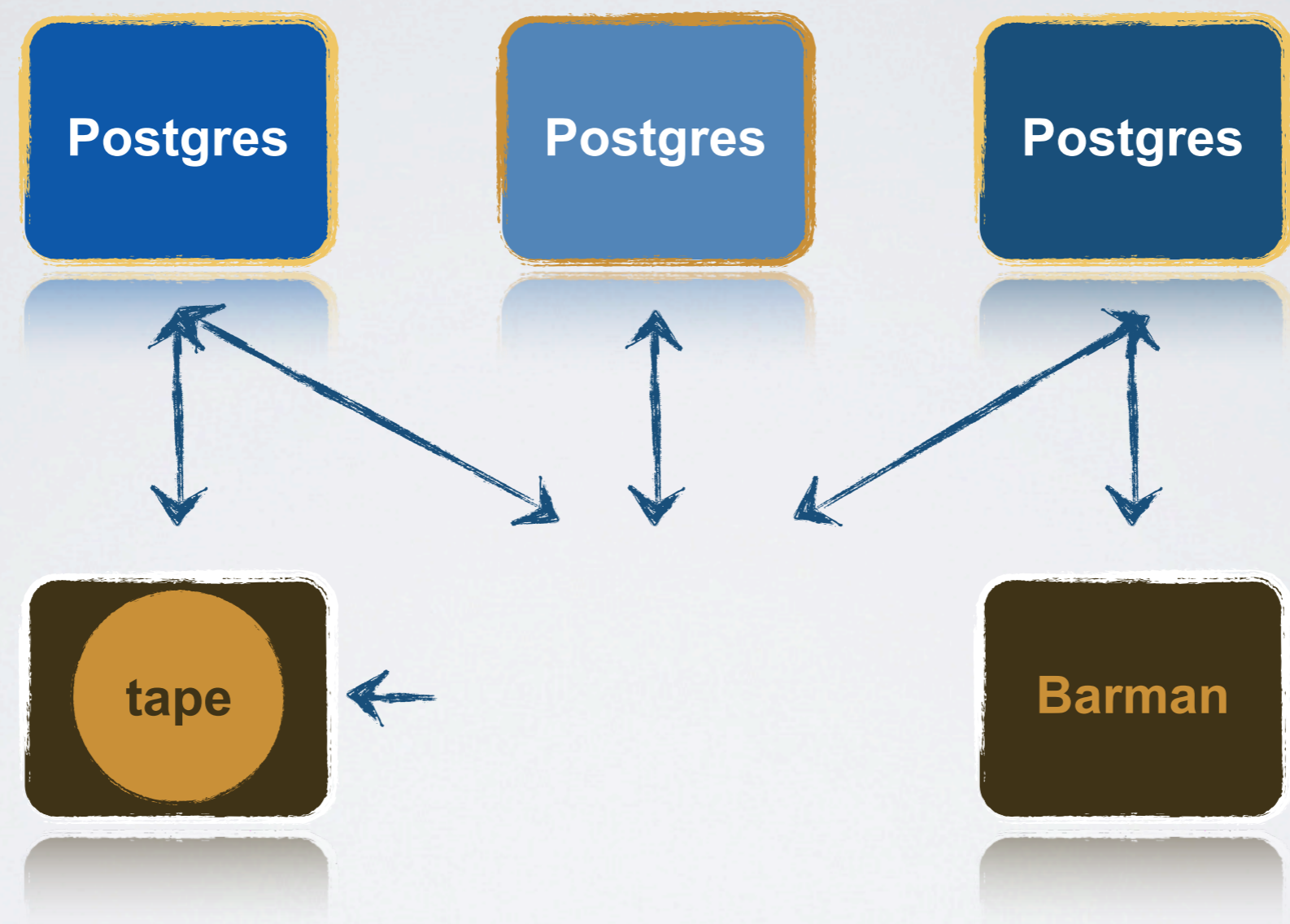
Backup and recovery
manager for PostgreSQL

PART II

Disaster recovery with Barman for PostgreSQL

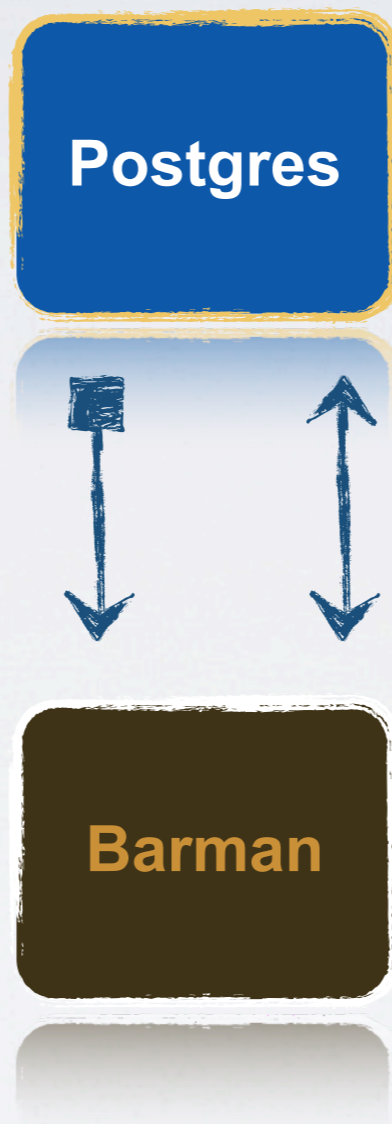
BARMAN

- GNU GPL 3
- Hosted on Sourceforge.net
- Python 2.6/2.7
- PostgreSQL 8.4, 9.0, 9.1, 9.2
- PyPI package
- RPM package
- Debian package (soon)
- Designed, developed, maintained by 2ndQuadrant

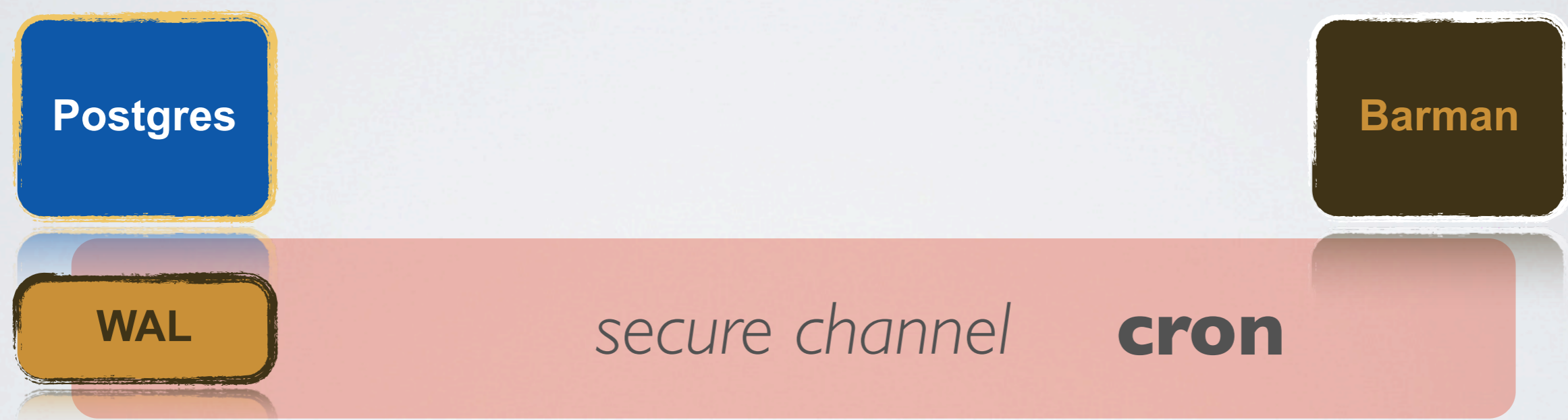


LAN, centralized architecture

Continuous archiving
(*WAL shipping via SSH*)



SSH commands
SQL commands



*Barman's
WAL archive*

Postgres

Periodical backup (weekly)

Differential backup

Backup catalogue

Barman

Full backup - Sat 1, 4AM

Full backup - Sat 8, 4AM

Full backup - Sat 15, 4AM

Full backup - Sat 22, 4AM

CONVENTION OVER CONFIGURATION

global/per server options
default directory layout

CONFIGURATION FILE

[barman]

```
barman_home = /srv/barman
barman_user = barman
log_file = /var/log/barman/barman.log
log_level = NOTICE
compression = gzip
```

[production]

```
description = Production PostgreSQL
ssh_command = ssh pg.2ndQuadrant.it
conninfo = host=pg.2ndQuadrant.it user=postgres
compression = bzip2
```

CONVENTIONAL DIRECTORIES FOR BARMAN

- **barman_home** (/srv/barman)
 - *server directory* (/srv/barman/production)
 - *base directory* (/srv/barman/production/base)
 - *WAL directory* (/srv/barman/production/wals)
 - *incoming directory* (/srv/barman/production/incoming)

MULTI-SERVER CONFIGURATION

```
[barman]
; General configuration
; ...
[server_one]
; Configuration for Server 1
; ...
[server_two]
; Configuration for Server 2
; ...
[server_x]
; ...
```

GLOBAL COMMANDS

- **List of managed servers**
 - barman list-server
- **Maintenance operations**
 - barman cron

SERVER COMMANDS

- **Information and diagnostics**

- barman status
- barman check
- barman show-server
- barman list-backup

- **Backup control**

- **Recovery control**


```
Terminal
[barman@backupserver]~$ barman list-backup test-9.1
test-9.1 20120821T164531 - Tue Aug 21 16:45:38 2012 - Size: 109.0 MiB - WAL Size: 304.0 MiB
test-9.1 20120821T163806 - Tue Aug 21 16:40:37 2012 - Size: 68.0 MiB - WAL Size: 208.0 MiB
test-9.1 20120821T163146 - Tue Aug 21 16:32:10 2012 - Size: 45.0 MiB - WAL Size: 432.0 MiB
test-9.1 20120821T162630 - Tue Aug 21 16:26:32 2012 - Size: 53.0 MiB - WAL Size: 400.0 MiB
test-9.1 20120821T161651 - Tue Aug 21 16:16:53 2012 - Size: 49.0 MiB - WAL Size: 48.0 MiB
[barman@backupserver]~$ █
```

BACKUP CONTROL

- barman backup
- barman show-backup
- barman list-files
 - standalone, data, wal, full
- barman delete

```
Terminal
[barman@backupserver]~$ barman backup test-9.1
Starting backup for server test-9.1 in /tmp/barman-tests/test-9.1/base/20120821T163235
Backup start at xlog location: 0/29000020 (00000001000000000000000029, 00000020)
Copying files.
Copy done.
Asking PostgreSQL server to finalize the backup.
Backup end at xlog location: 0/290000A0 (00000001000000000000000029, 000000A0)
Backup completed

[barman@backupserver]~$ █
```

SHOW BACKUP

- **General**

- Server name, Postgres version, status, ...

- **Base backup**

- Start/End time, first/last WAL, disk usage, ...

- **WAL**

- Number of associated files
- disk usage

- **Context**

- Previous/Next backup

```
Terminal
[barman@backupserver]~$ barman show-backup test-9.1 20120821T161651
Backup 20120821T161651:
Server Name      : test-9.1
Status:         : DONE
PostgreSQL Version: 90104
PGDATA directory : /tmp/barman-tests/9.1/master-node/data

Base backup information:
Disk usage      : 65.0 MiB
Timeline       : 1
Begin WAL      : 00000001000000000000000000000000C
End WAL        : 00000001000000000000000000000000C
WAL number     : 1
Begin time     : 2012-08-21 16:16:51.303886
End time       : 2012-08-21 16:16:53.243259
Begin Offset   : 32
End Offset     : 160
Begin XLOG     : 0/C000020
End XLOG       : 0/C0000A0

WAL information:
No of files    : 3
Disk usage     : 48.0 MiB
Last available : 00000001000000000000000000000000F

Catalog information:
Previous Backup : - (this is the oldest base backup)
Next Backup    : 20120821T162630
[barman@backupserver]~$ █
```

RECOVERY CONTROL

- **Recovery target** (full / point in time)
- **Local recovery**
 - `barman recover`
- **Remote recovery**
 - `barman recover --remote-ssh-command`

ADVANCED RECOVERY

- **Point In Time Recovery**

- `--target-time = TIME`
- `--target-xid = XID`

- **Relocation of tablespaces**

- `--tablespace NAME:LOCATION [...]`

COMMON USE CASES

- Accidental errors recovery
- Disaster recovery
- Sandbox server (BI, staging, ...)


```
Terminal
[barman@backupserver]~$ barman recover test-9.1 20120821T161651 /srv/postgres/recover/
Starting local restore for server test-9.1 using backup 20120821T161651
Destination directory: /srv/postgres/recover/
Copying the base backup.
Copying required wal segments.
The archive_command was set to 'false' to prevent data losses.
Restore done!

Please review the network and archiving related settings
in the postgres configuration file before start the just recovered instance.

[barman@backupserver]~$ █
```

SPONSOR A FEATURE!

- Retention policies
- Sandbox recovery
- Replication protocol support (client / server)
- Incremental backup
 - reduce backup time and size

RETENTION POLICY CONFIGURATION

```
; Base backup retention policy  
retention_policy = 'redundancy 3'  
retention_policy = 'recovery window of 30 days'
```

```
; WAL retention policy  
wal_retention_policy = 'base' ;  
wal_retention_policy =  
    'recovery window of 14 days'
```

OUR COMMITMENT

- Keep it open source
- Reinvest money from sale of DR turnkey solutions in R&D
- Support and maintain RPM/Debian packages
- Accept sponsorships for new features development

CONCLUSIONS

- Hides complexity of PITR / Keeps unaltered PITR strenghts
- Not invasive
- Fosters migrations from Oracle
- “*Standard de facto*” for PostgreSQL Disaster Recovery
- Advice: plan for DR (if you have not done it yet)



Barman
Backup and recovery
manager for PostgreSQL

QUESTIONS?

Gabriele.Bartolini@2ndQuadrant.it

Twitter: [_GBartolini_](#)

www.pgbarman.org

www.2ndQuadrant.com

2ndQuadrant 
Professional PostgreSQL

MERCI BEAUCOUP!

LICENSE



This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License

Copyright (c) 2012 - 2ndQuadrant.it