Continuità operativa di database PostgreSQL

Gabriele Bartolini
2ndQuadrant Italia / ITPUG
gabriele.bartolini@2ndQuadrant.it
@_GBartolini_

Gabriele Bartolini

- Co-Fondatore e Manager di 2ndQuadrant Italia
  - Data Architect in Ambienti business critical
  - Data warehousing
- Co-Fondatore Italian PostgreSQL Users Group
- Co-Fondatore PostgreSQL Europe
- Attivista comunità PostgreSQL

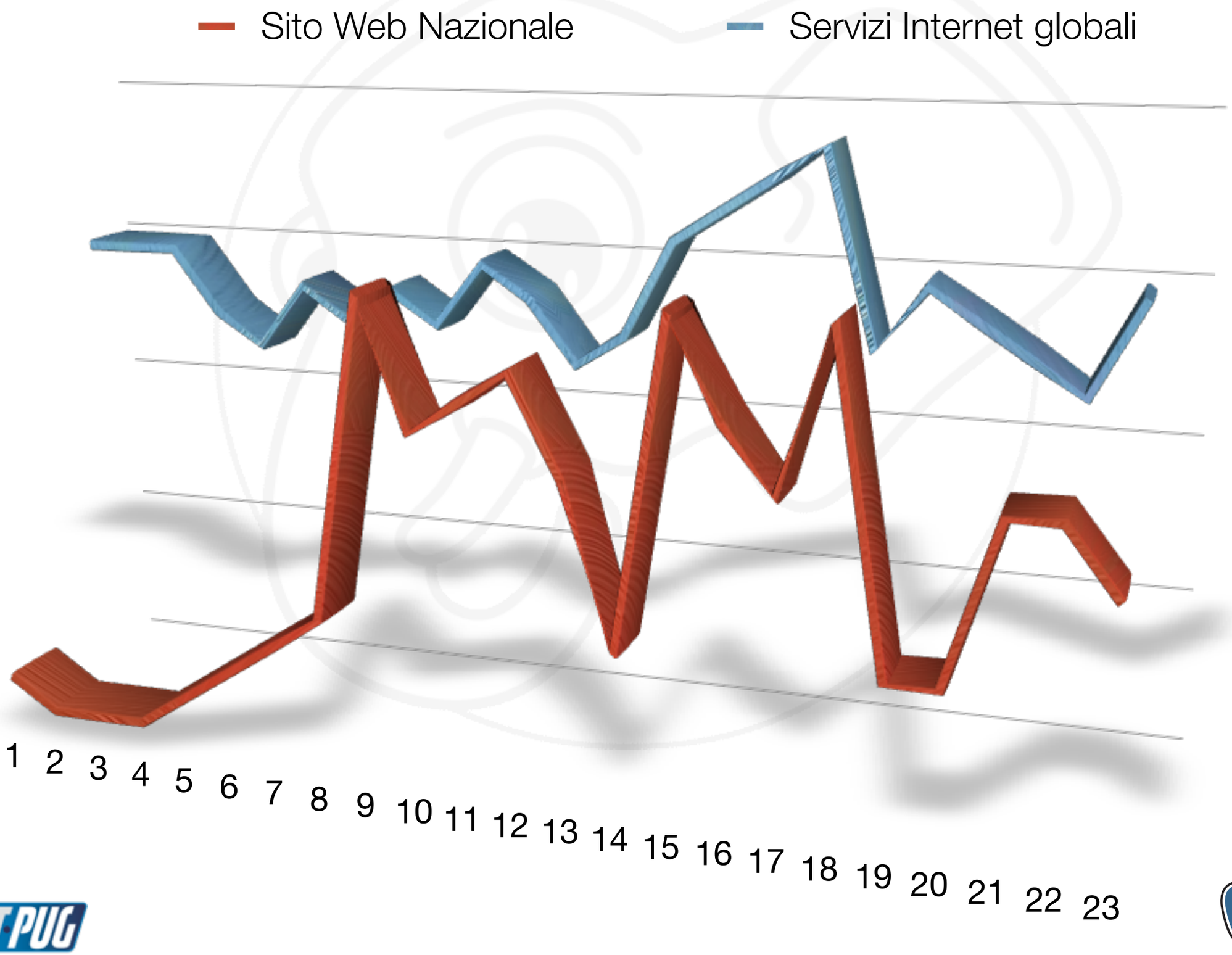
Sommario

- Panoramica su continuità operativa
- Continuità operativa su PostgreSQL 9
- Disaster recovery con PostgreSQL 9
- Alta disponibilità con PostgreSQL 9
- Conclusioni

Parte I

Una panoramica sulla continuità operativa

Business 24/7/365



IT PUG

PostgreSQL

Business Continuity

In italiano = continuità operativa

Definizione nel CAD

*"Insieme di **attività** volte a **ripristinare** lo stato del sistema informatico o parte di esso, **compresi gli aspetti fisici e organizzativi e le persone necessarie per il suo funzionamento**, con l'obiettivo di riportarlo alle condizioni antecedenti a un **evento disastroso**"*

Decreto legislativo n. 82/2005 e successive modifiche, Art. 50-bis "Continuità operativa"

IT-PUG

PostgreSQL

Business Continuity in ICT

- **Alta disponibilità**
  - *High Availability*
- **Disaster recovery**
- *Torneremo più tardi su questi due argomenti*

Parte II

Continuità operativa di database PostgreSQL 9

Evento avverso

Guasto al sistema

Errore umano accidentale #1

kill -9

Postgres Backend Process

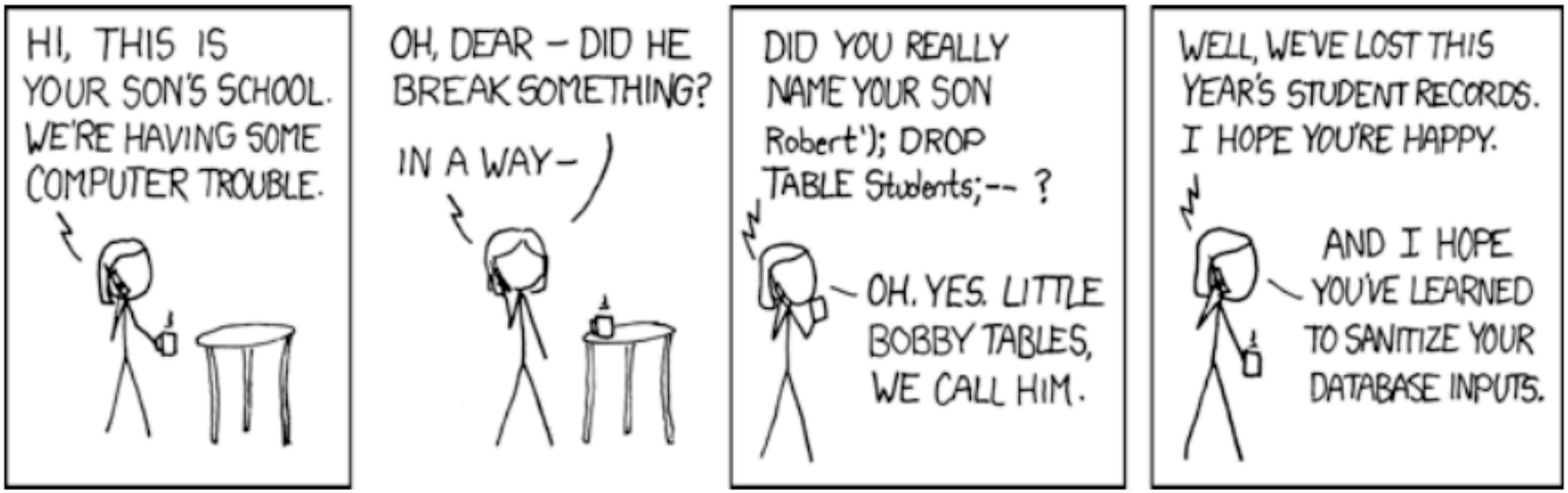
Errore umano accidentale #2

```
postgres@pg $> psql
psql (9.2.1)
Type "help" for help.

postgres=# DROP DATABASE d;
DROP DATABASE
postgres=# \q
postgres@pg $> date
Ven 16 Nov 2012 18:12:47 CET
```

Errore umano accidentale #3

```
postgres@pg $> psql
psql (9.2.1)
Type "help" for help.

postgres=# UPDATE t SET colore = ‘giallo’;
UPDATE 4815162342
postgres=# ^C
^C
^C
^Z
```

“Little” Bobby Tables



IT PUG

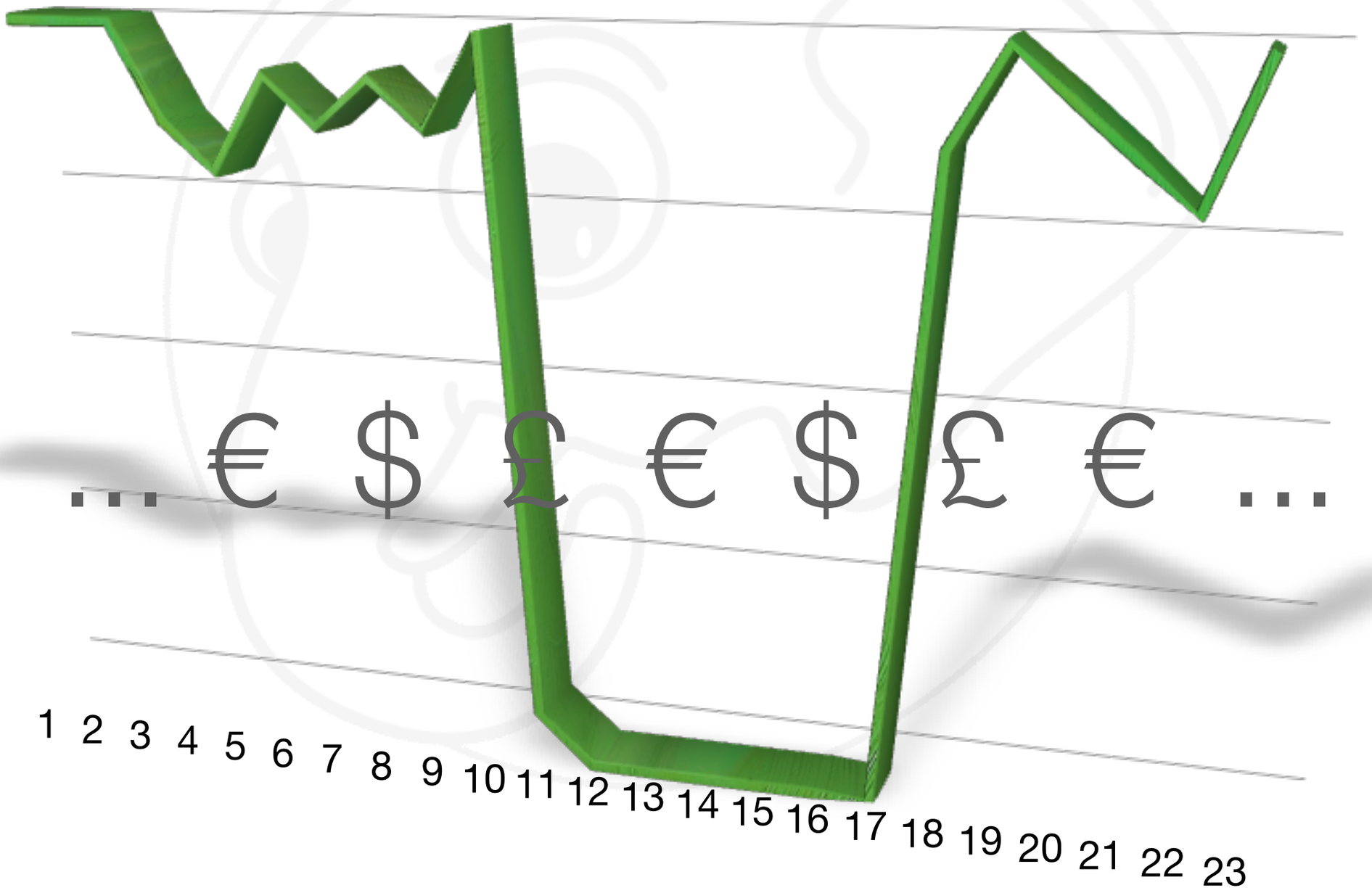
PostgreSQL

Disastro

Conseguenze di un evento avverso

- **Disservizio**
  - *Down time*
- **Perdita di dati**
  - *Data loss*

Downtime



IT PUG

PostgreSQL

Indicatori di continuità operativa

- Recovery **Point** Objective
  - *Quanti e quali dati ripristinare in seguito a un evento avverso?*
  - **RPO**
- Recovery **Time** Objective
  - *In quanto tempo ripristinare la continuità operativa?*
  - **RTO**

Perdita dati 0

- Zero data loss
- **RPO = 0**
- PostgreSQL 9.1 introduce replica sincrona in streaming

Downtime 0

- Uptime 100%
  - **RTO = 0**
  - Utopia
- **RTO ~ 0** con uptime plausibili:
  - 99.95% (~ 4hr/anno)
  - 99.99% (~ 1hr/anno)
  - 99.999% (~5m/anno)

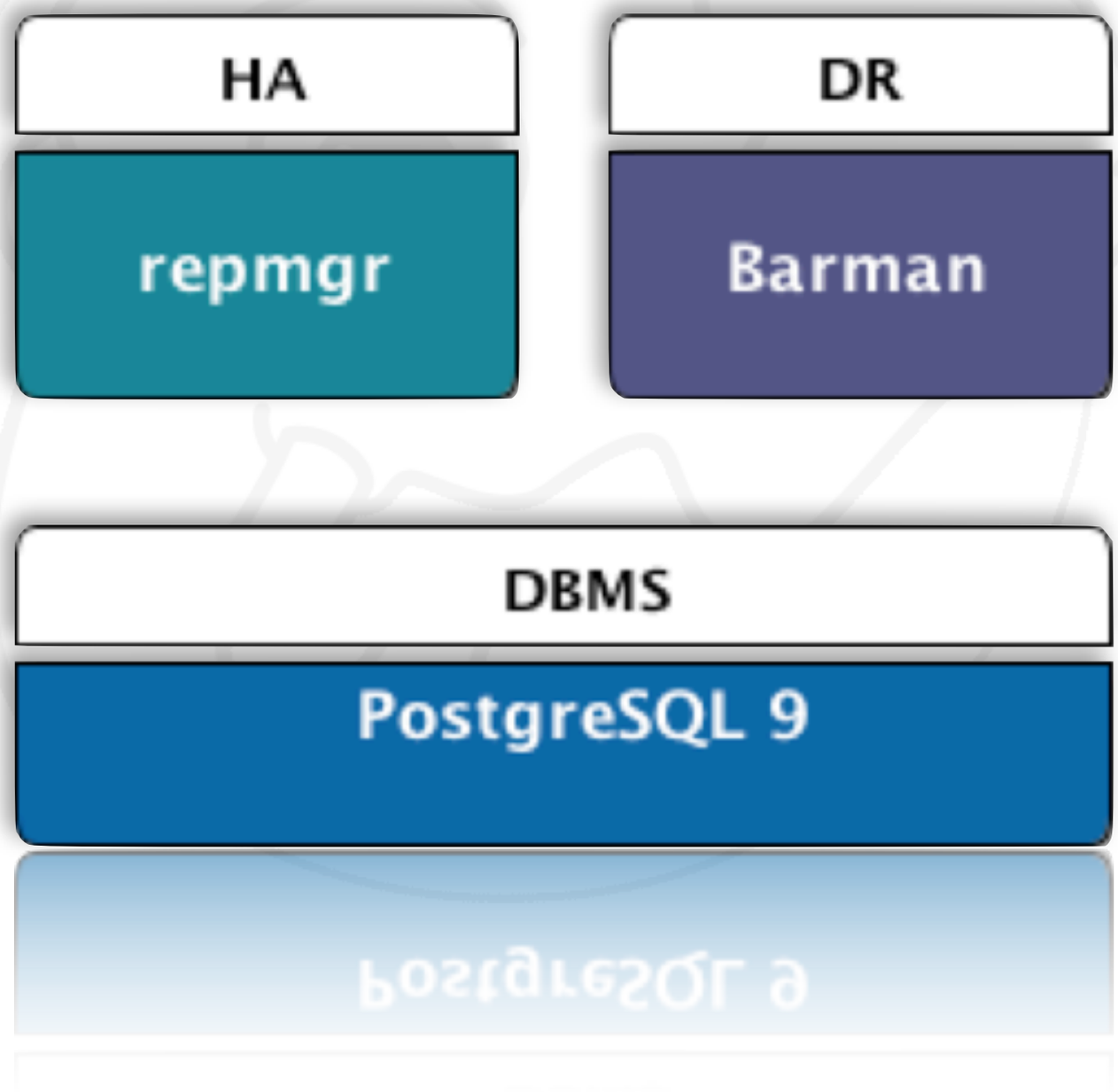
Obiettivi

- Ideale:
  - RPO = 0, RTO < 1 minuto
  - Costi elevati (procedure completamente automatiche)
- Solitamente:
  - **RPO < 5 minuti di transazioni, RTO < 1 ora**
  - Investimento in prevenzione e pianificazione

Caratteristiche di PostgreSQL 9

- Disaster Recovery
  - Backup logico con pg_dump e pg_restore
  - Backup fisico e Point In Time Recovery
- Alta disponibilità
  - Replica asincrona con log shipping
  - Replica in streaming asincrona (o sincrona da 9.1)

Stack open source PostgreSQL



Parte III

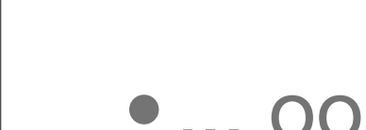
Disaster Recovery con PostgreSQL 9

Disaster Recovery

- Pianificare e predisporre misure:
  - organizzative (procedure)
  - tecnologiche
    - hardware e software, rete, memorie di massa, …
- Reagire ad un disastro
  - Backup e Restore

Caso #0: nessun backup

- ...
- ... mhhhh ...
- ... ooops ...



Caso #1: RPO di un “dump”

- **Backup logico a caldo**
  - pg_dump -Fc per ogni database
  - pg_dumpall -g
- Backup periodico (snapshot)
  - giornaliero: RPO < (24 ore + tempo di backup)
  - settimanale: RPO < (168 ore + tempo di backup)

Caso #1: RTO di un “dump”

- Restore di un precedente “dump”/backup
  - pg_restore
- RTO = somma dei tempi per:
  - reinstallazione del sistema operativo (eventuale)
  - reinstallazione di PostgreSQL (eventuale)
  - **restore del database** o di un set di oggetti (e.g. tabelle**)**

04:00:01

t inizio

05:12:44

t fine

pg_dump -Fc

14 ore 43 minuti 14 secondi
di potenziale perdita dati

18:43:15

CRASH!!!

transazioni

t t t t t t t t t t t t t

Caso #1: riepilogo

- Contro:
  - Alto RPO
  - Alto RTO
  - **Spesso insufficiente per ambienti in continuità operativa**
- Pro:
  - Restore selettivo e parziale

Caso #2: RPO di PITR

- **Point In Time Recovery** (8.1)
- **Backup fisico a caldo completo**
  - pg_start_backup + copia fisica + pg_stop_backup
- **Backup differenziale**
  - WAL shipping (Continuous Archiving)
- **RPO configurabile** (e.g. < 5 minuti)

Caso #2: RTO di PITR

- Restore di un precedente backup e archivio WAL
- RTO = somma dei tempi per:
  - reinstallazione del sistema operativo (eventuale)
  - reinstallazione di PostgreSQL (eventuale)
  - **copia fisica dei backup**
  - **replay dei segmenti WAL**

04:00:01

t inizio

04:41:52

t fine

Base Backup

1 minuto 1 secondi
di potenziale perdita dati

18:43:15

transazioni

CRASH!!!

WAL WAL WAL WAL WAL WAL WAL WAL WAL WAL WAL WAL

18:42:14

IT PUG

PostgreSQL

Caso #2: riepilogo

- Contro:
  - **No ripristino selettivo** (intera istanza Postgres)
  - **Primitive di PostgreSQL complesse**
- Pro:
  - Robusto, affidabile, flessibile
  - Migliori RPO e RTO

Barman

- Software Open Source per Backup e Recovery
  - GPL 3
  - Luglio 2012
- Si basa su PITR
- Semplifica e migliora la gestione di backup e recovery
- Sito: www.pgbarman.org

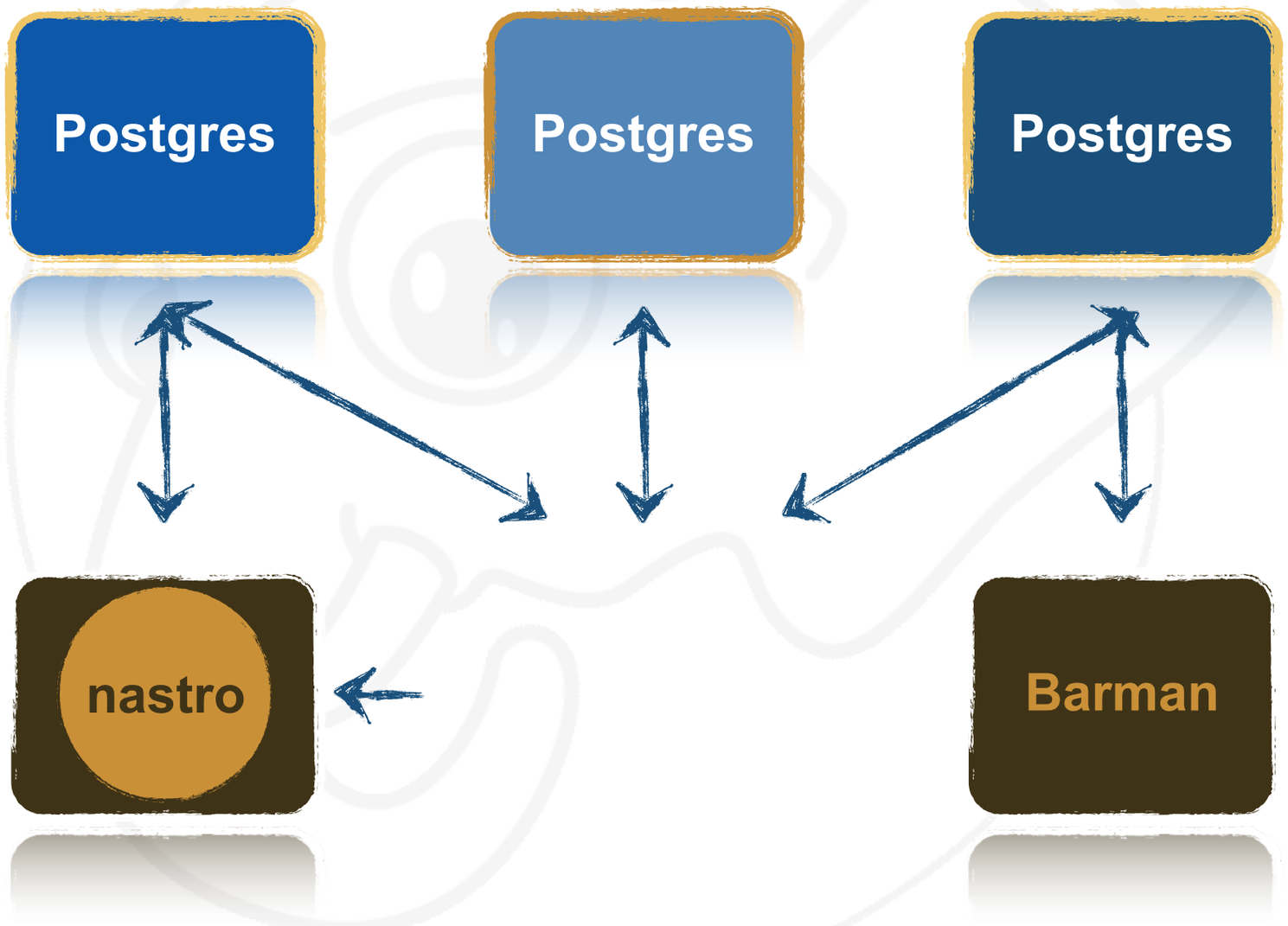
Barman
Backup and recovery
manager for PostgreSQL

IT PUG

PostgreSQL

Obiettivi di Barman

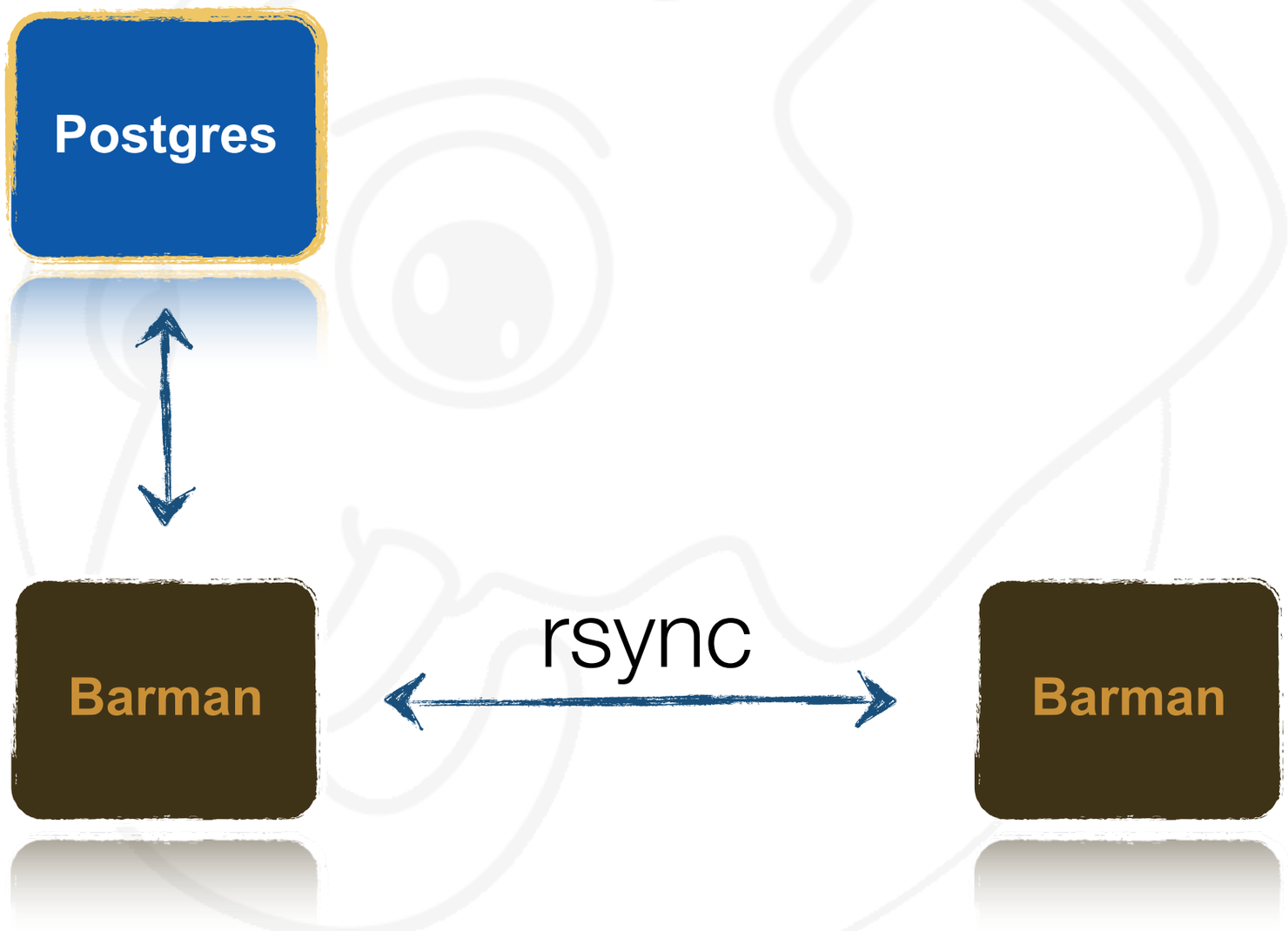
- Backup a caldo, full, incrementale e differenziale
- Server multipli
- Backup e PITR remoti
- Cataloghi di backup
- Retention policy
- Archiviazione/compressione
  - segmenti WAL
  - backup periodici
- **Automazione**
- **Integrazione**
- **Usabilità**



LAN, architettura ibrida

data centre 1

data centre 2



Ridondanza geografica

Barman
Backup and recovery manager for PostgreSQL

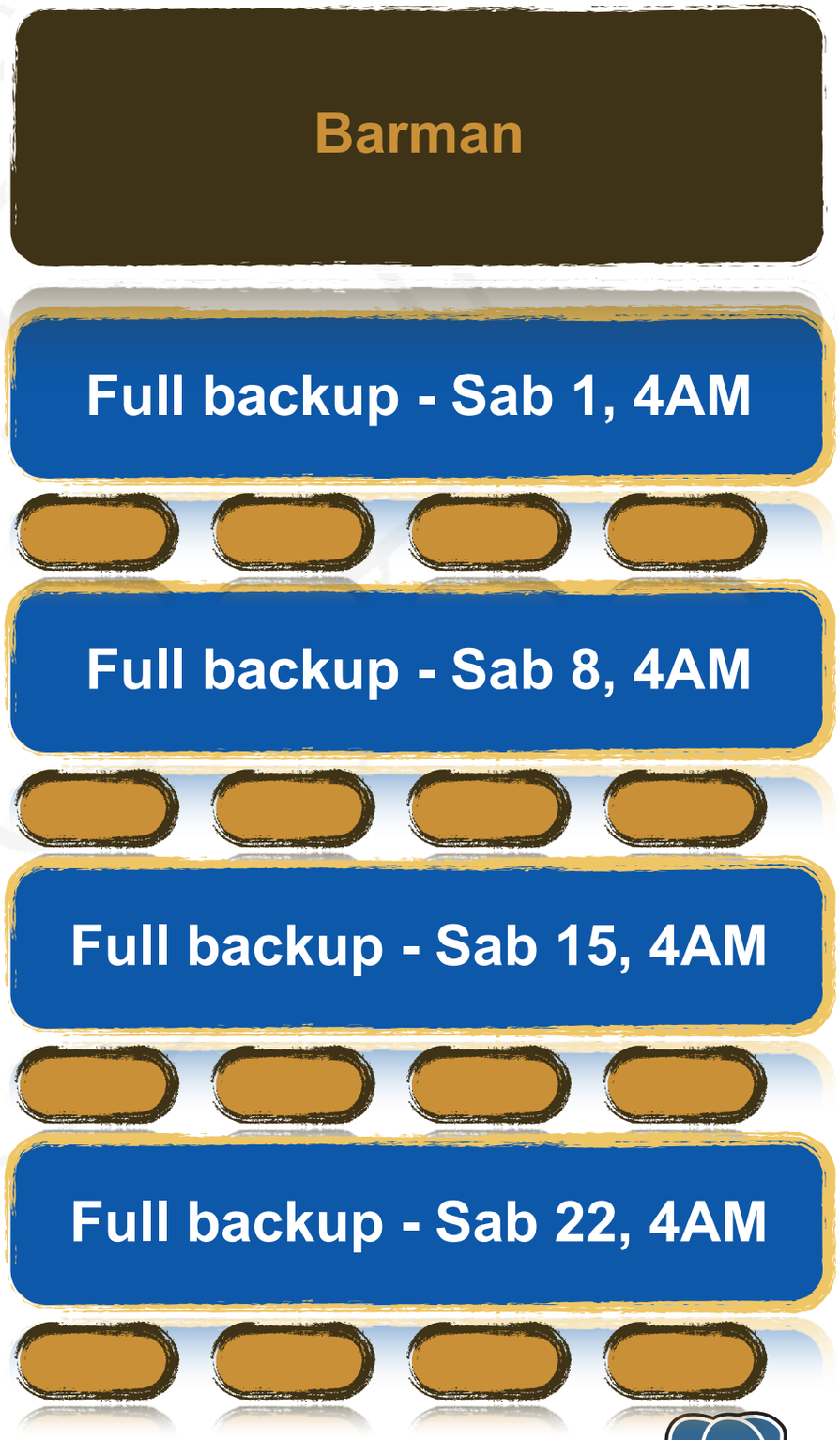
Postgres

Barman

Backup periodico (settimanale)

Backup differenziale

Catalogo dei backup



IT PUG

PostgreSQL

```
$ barman show-backup main 20121110T130001

Backup 20121110T130001:
  Server Name        : main
  Status:            : DONE
  PostgreSQL Version: 80408
  PGDATA directory   : /srv/postgresql/main/data
  Tablespaces:
    tb_data1: /srv/tablespace/tb_data1
    tb_temp : /srv/tablespace/tb_tmp

  Base backup information:
    Disk usage       : 6.0 TiB
    Timeline         : 1
    Begin WAL        : 00000001000386D00000042
    End WAL          : 000000010000387000000090
    WAL number       : 844
    Begin time       : 2012-11-10 13:00:01.839077
    End time         : 2012-11-11 10:34:32.722517
    Begin Offset     : 39832
    End Offset       : 13773184
    Begin XLOG       : 386D/42009B98
    End XLOG         : 3870/90D22980

  WAL information:
    No of files      : 34751
    Disk usage       : 179.0 GiB
```

2ndQuadrant
Professional PostgreSQL

www.pgbarman.org

Barman

Backup and recovery
manager for PostgreSQL

IT PUG

PostgreSQL

Parte IV

Alta disponibilità con PostgreSQL 9

Privilegiare RTO

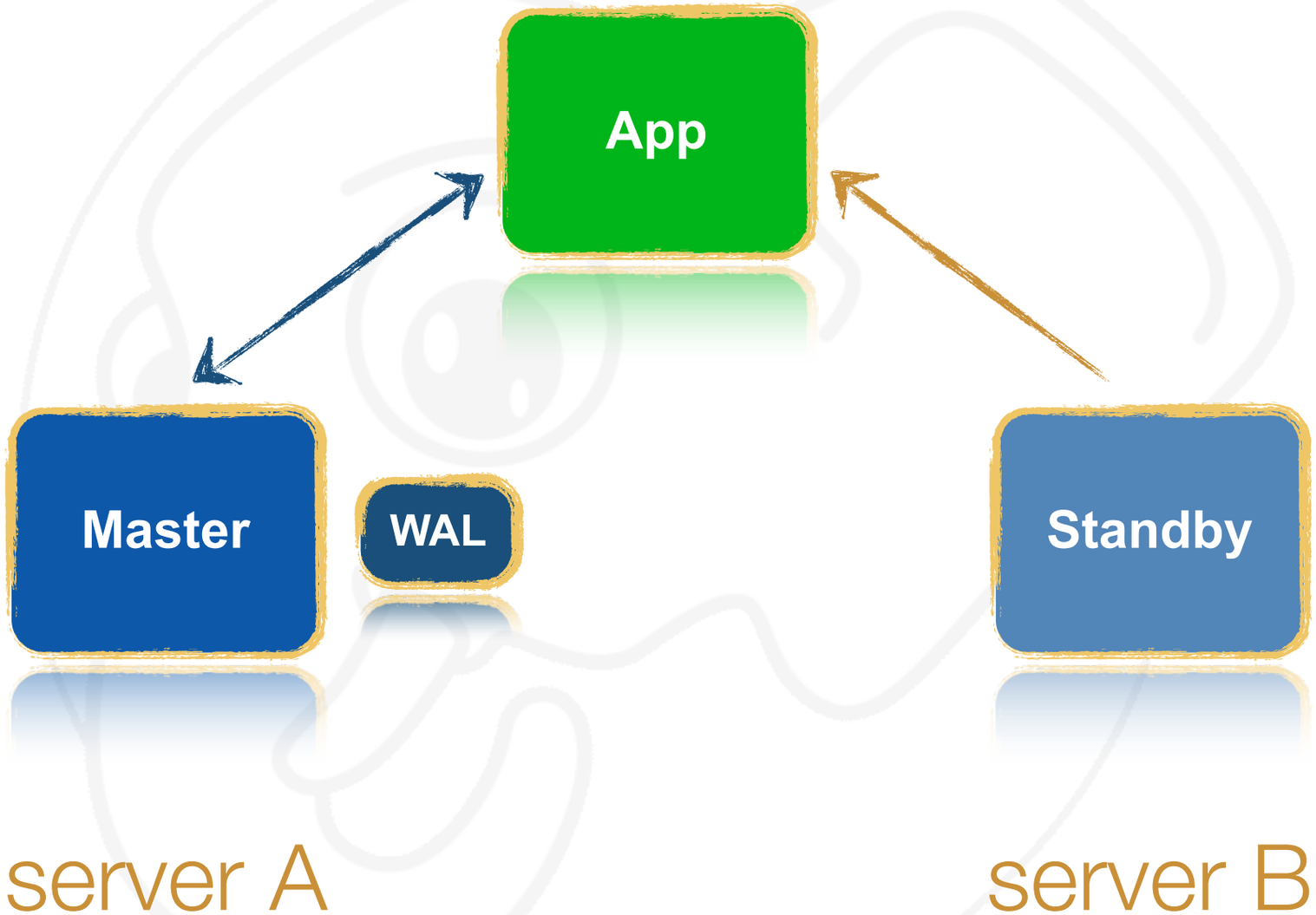
- Disaster recovery:
  - privilegia RPO
  - prima forma di alta disponibilità (ad alto RTO)
- Esigenze di RTO basso:
  - **Alta disponibilità**
  - **Replica master/slave**

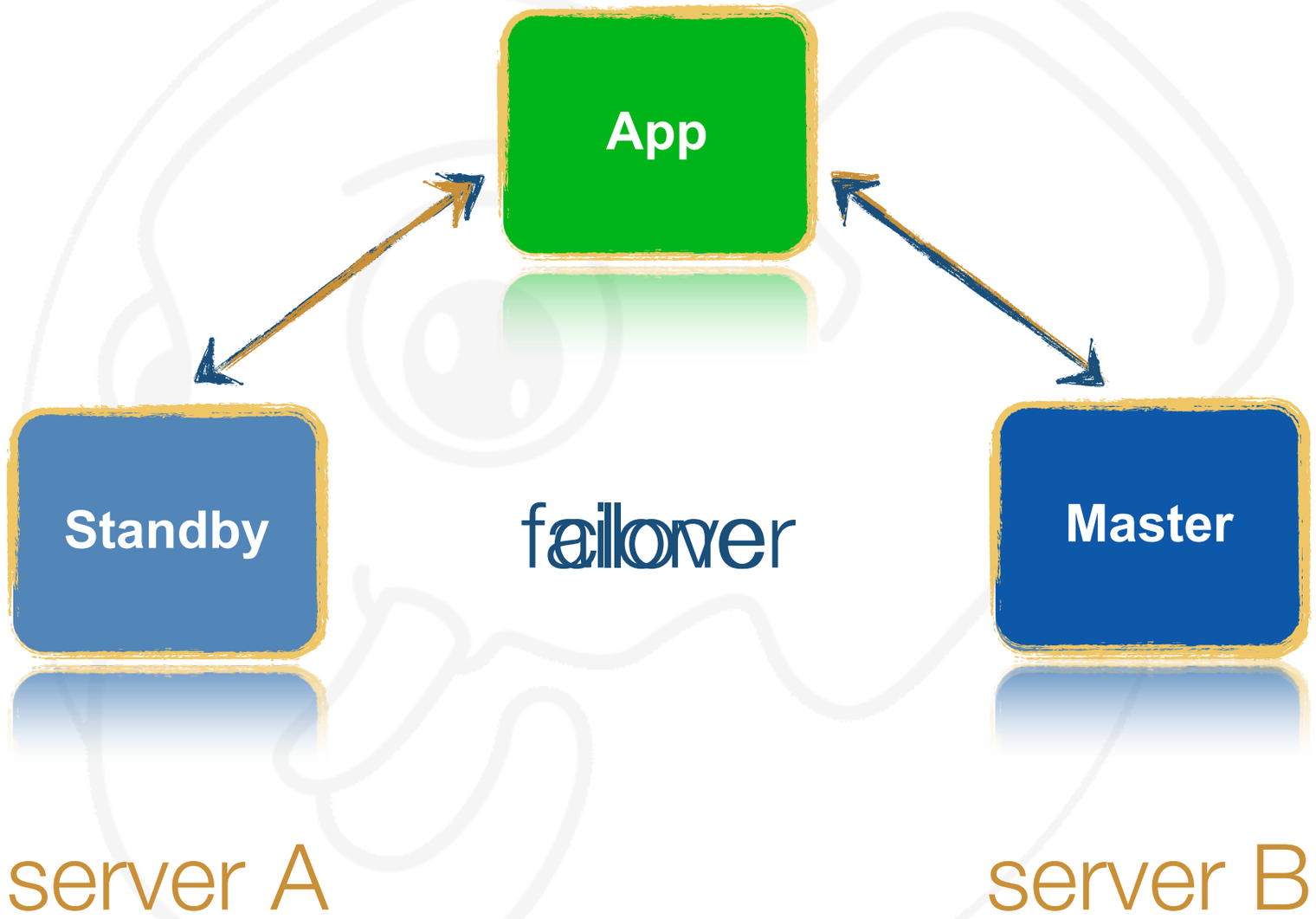
Log shipping

- PostgreSQL produce regolarmente file WAL
  - crash recovery
  - 16MB
- In modalità “*continuous archiving*”:
  - archivia i file WAL (archive_command)
  - asincronia controllata (archive_timeout)

Streaming replication

- A partire da 9.0
- Utilizza il protocollo di replica di PostgreSQL
- Processo WAL receiver sullo standby
  - Processo WAL sender sul master
- Permette replica sincrona
  - controllabile a livello di transazione!


App
Master
WAL
Standby
server A
server B


App
Standby
failover
Master
server A
server B

Failover

- Manuale
- Automatico
  - Attenzione a “split brain”

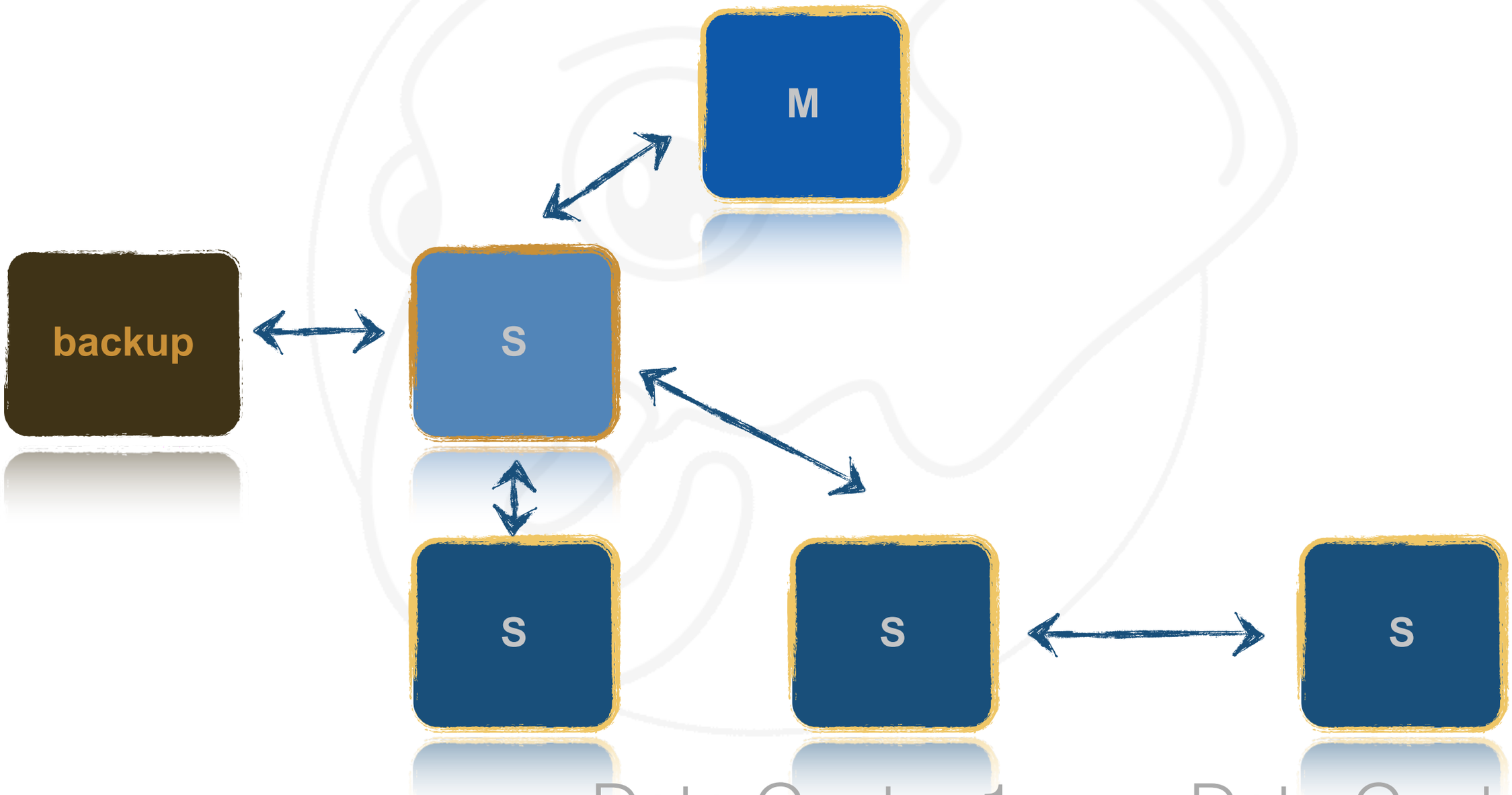
Strumenti open source

- repmgr
  - anche failover manuale
  - non invasivo
  - www.repmgr.org
- Pacemaker (Cluster Resource Manager)
  - www.clusterlabs.org
- pgpool
  - www.pgpool.net

Parte V

Conclusioni

Libero di crescere



IT PUG

PostgreSQL

Un passo per volta!

- Pianificare e progettare
  - HA simmetrica
- Analisi dei costi
  - servono almeno 3 server nel data centre principale solo per Postgres
  - Valutare ridondanza geografica
- Disaster Recovery = forma primordiale di Alta Disponibilità
  - DR senza HA = OK
  - HA senza DR = NO!
- Formazione e test/ simulazioni periodiche!

Domande

E-mail: gabriele.bartolini@2ndQuadrant.it

Twitter: @_GBartolini_

Grazie!