pglogical & disaster recovery
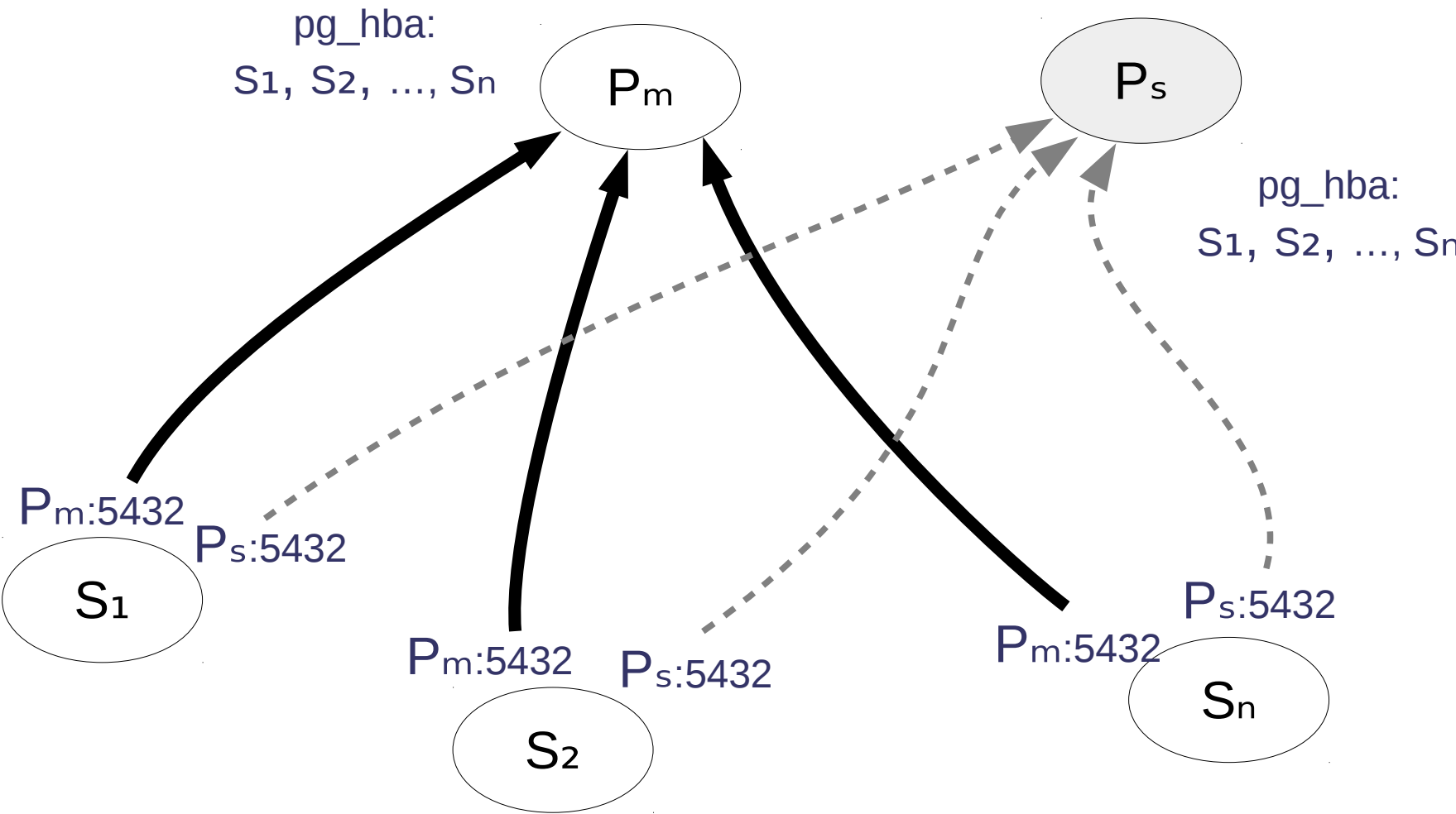
giovanni fabris – icon s.r.l.

pglogical in action

100÷1000 nodes network

- central provider (lin)
    - configuration
    - operational state
    - application heartbeat
    - may failover onto standbys
- network service (win/lin)
    - private DB
    - subscribes to (current) central provider

failover: recreate subscriptions

needs

transparent failover

- publisher fails
- subscribers detach
- publisher recovers on a standby
- subscribers reattach to new publisher
- no (distributed) intervention required

firewall bypass

- connections dropped on publisher…
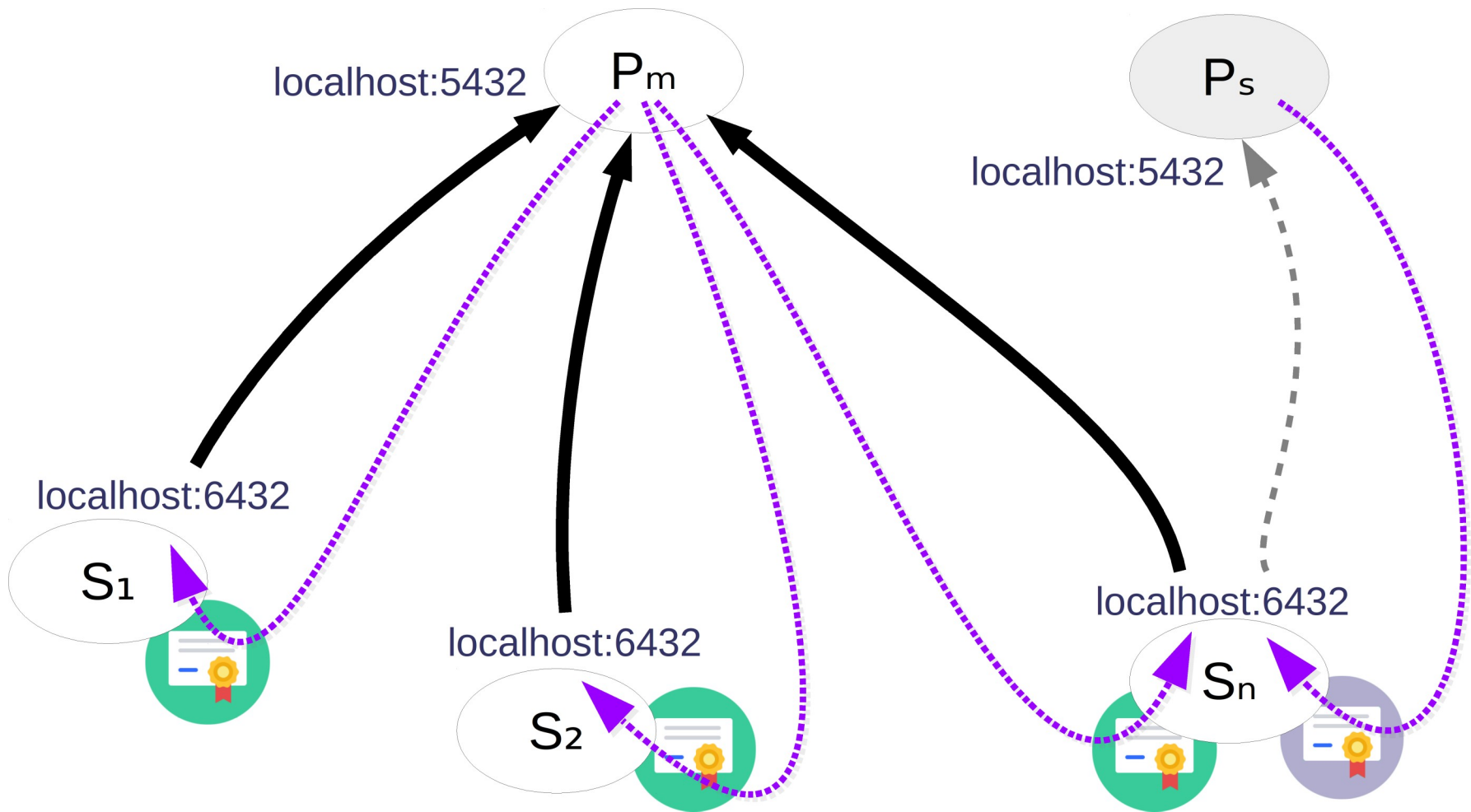- … but subscribers still waiting for replica

solution (?)

openssh on subscribers
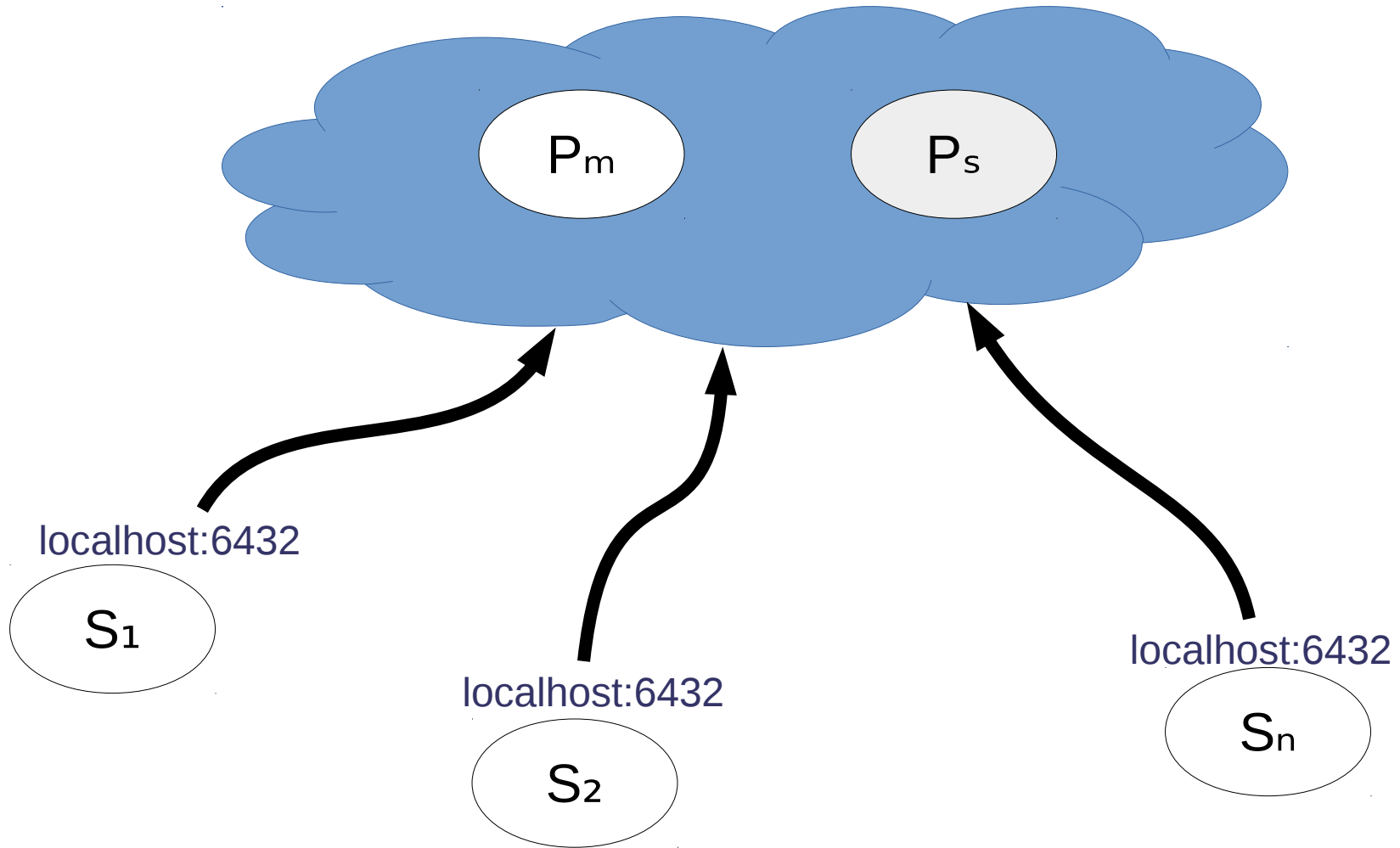
- lin – standard

- win – cygwin package

provider running reverse ssh tunnels

- public key authentication only

- virtualize subscriber's access to provider

- dropped with provider (or easily dropped by hand)

- recreated by newly promoted provider

reverse tunneling

localhost:5432

P_m

P_s

localhost:5432

localhost:6432

S_1

localhost:6432

S_2

localhost:6432

S_n

subscribers & providers on ssh

howto

ssh

    -fNT   # bg, no cmd, no tty

    -R 6432:localhost:5432   # reverse tunnel

    -o PasswordAuthentication=no # just pk auth

    -o TCPKeepAlive=yes   # byebye fw

    -o ControlMaster=yes   # enable control socket

    -S ~/rssh/%r@%h   # control master file

    DOM+ssh_usr@subscriber   # cygwin subscriber

the end

the good: side benefits

- channel encryption
- provider with localhost-only authentication

the bad: provider processes

- one process per subscriber
- cascade replication?

the ugly: will all this work?

- replication slots not shipped to standbys
- subscribers still need to resynch to new master ☹
- but pg11…