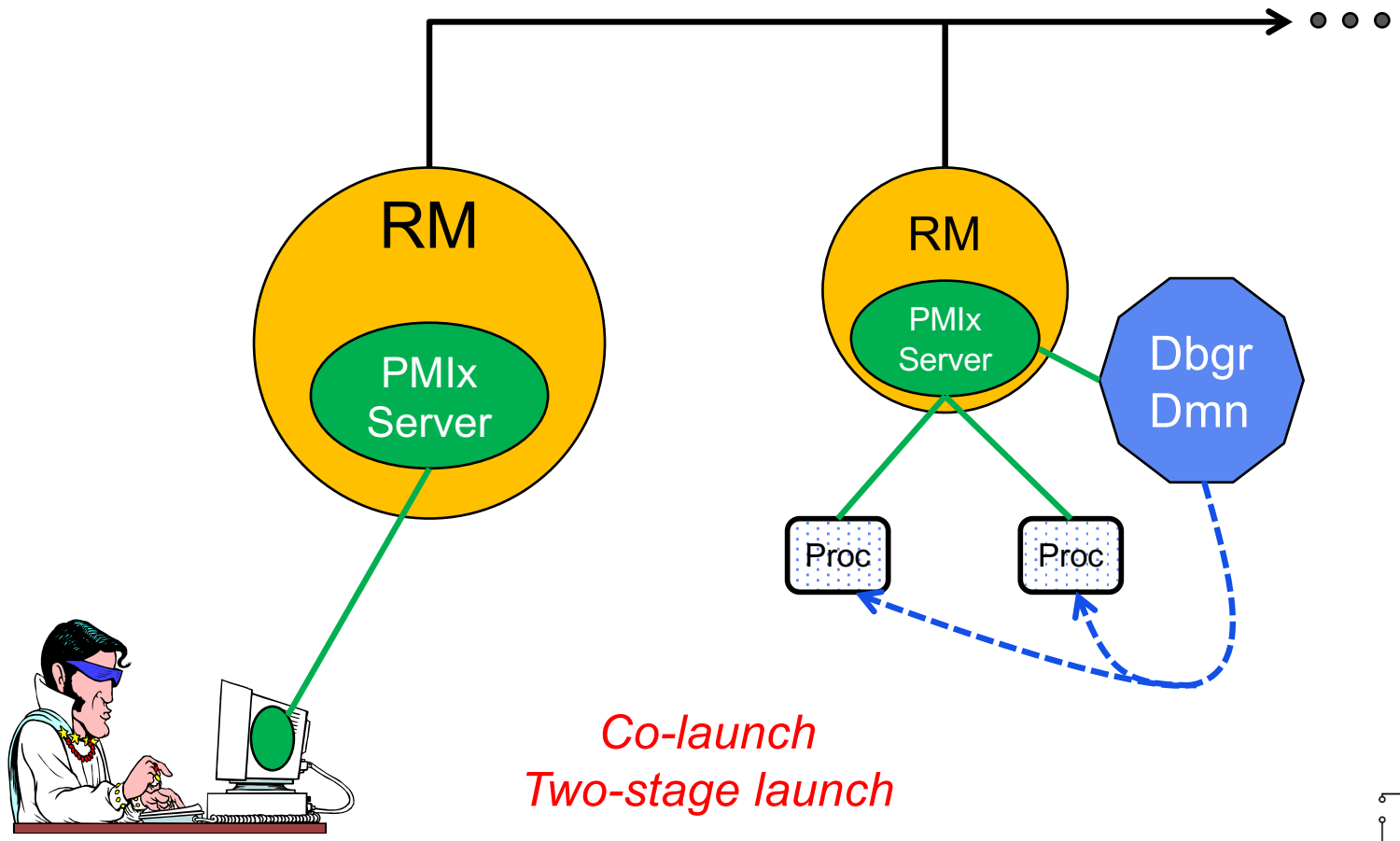
PMIx: Debuggers and Fabric

Debugger/Tool Features

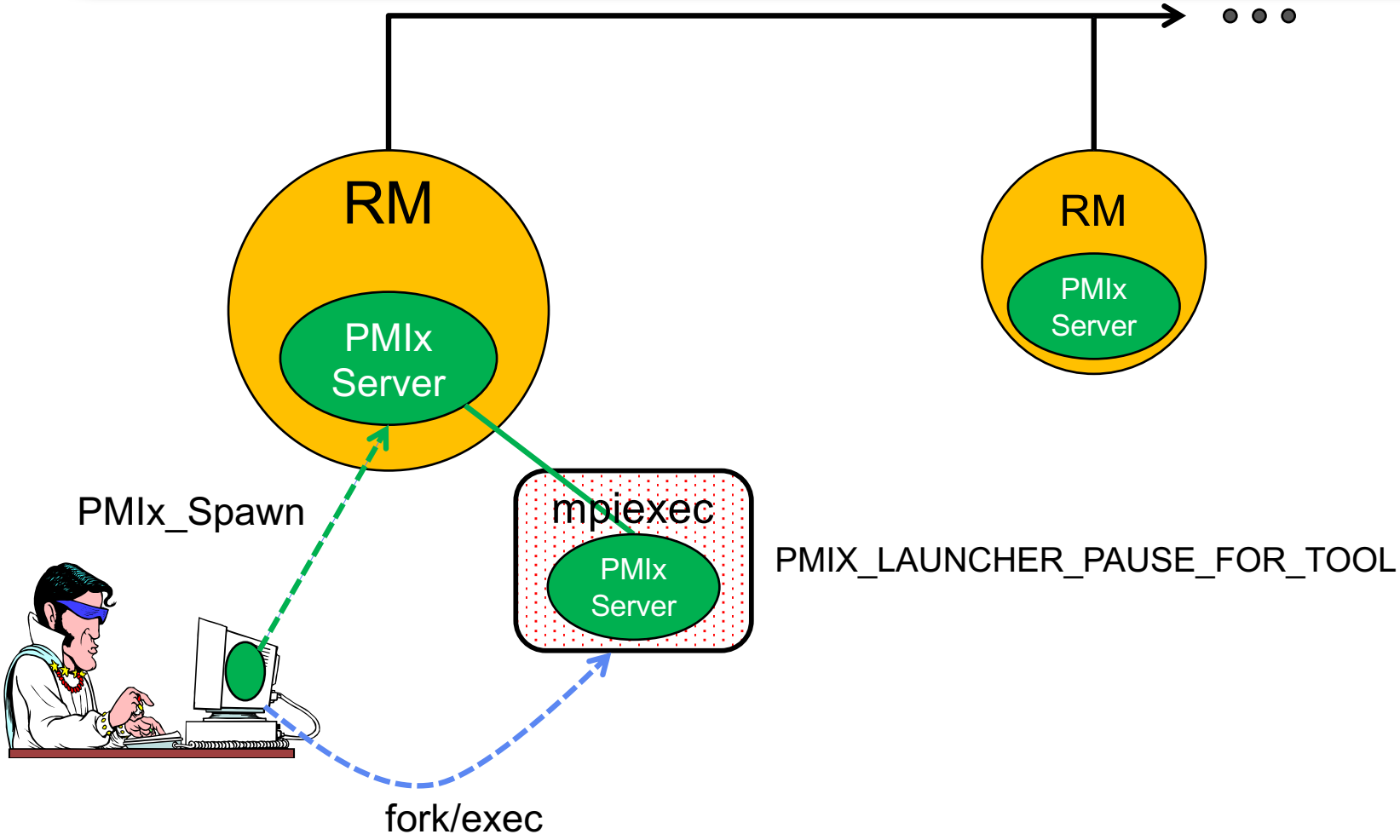
- Co-launch/co-location of daemons
  - At initial app spawn
    - Co-launch
  - Upon attach
    - Spawn w/co-location
- Launch control
  - Stop-on-exec, stop-in-init, stop-in-app
  - Release method spec
- Forwarding of IO
  - To/from debugger daemons
  - To/from app being debugged
- Query support
  - Global and local proctable
  - PMIx server and application procs
- Direct/indirect launch support
  - Forward, set/unset/modify envars (e.g., LD_PRELOAD)
  - Launcher directives
    - Modify local fork/exec agent
    - Replace launcher daemons

Direct Launch

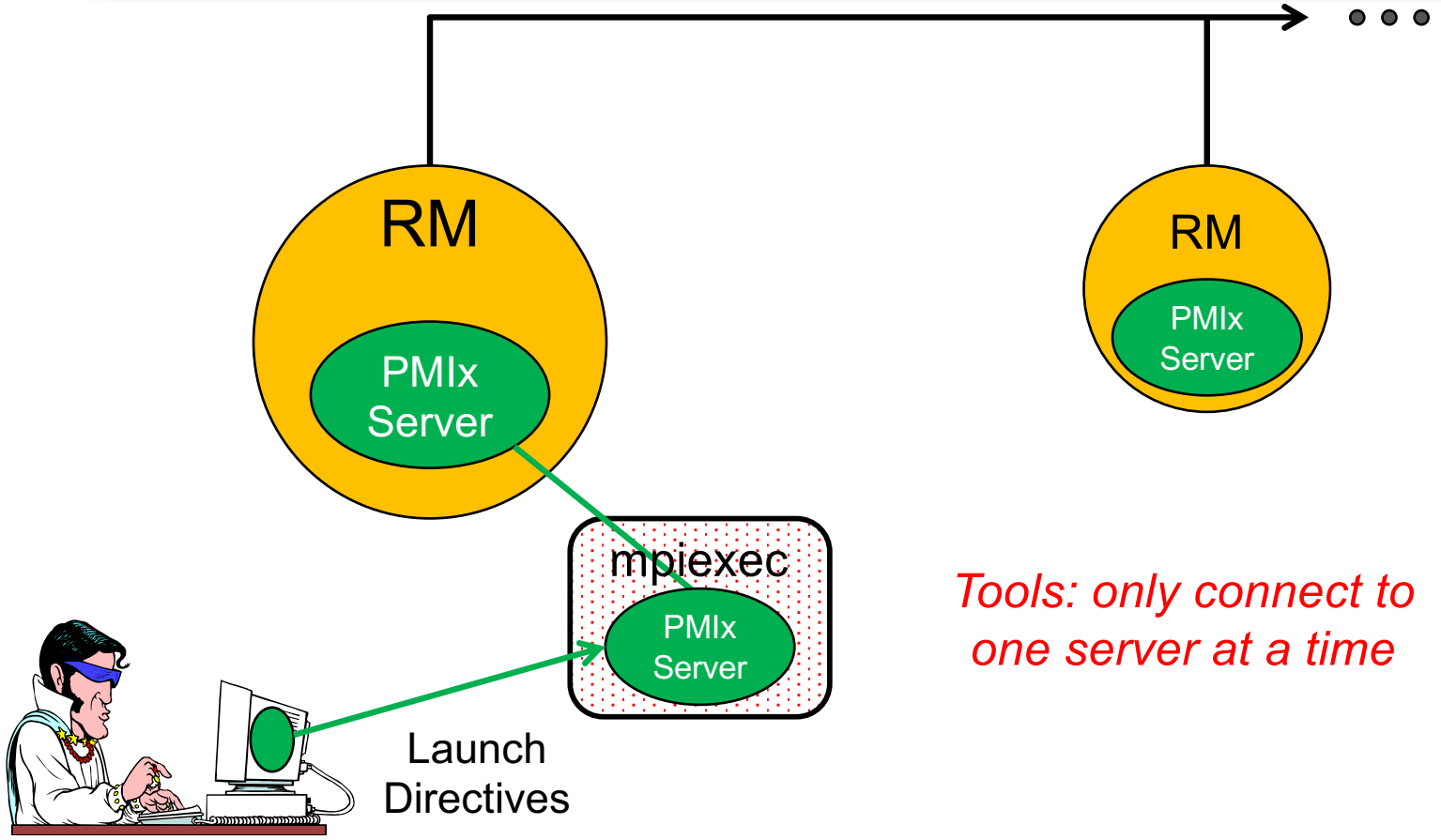


Co-launch
Two-stage launch

Indirect Launch

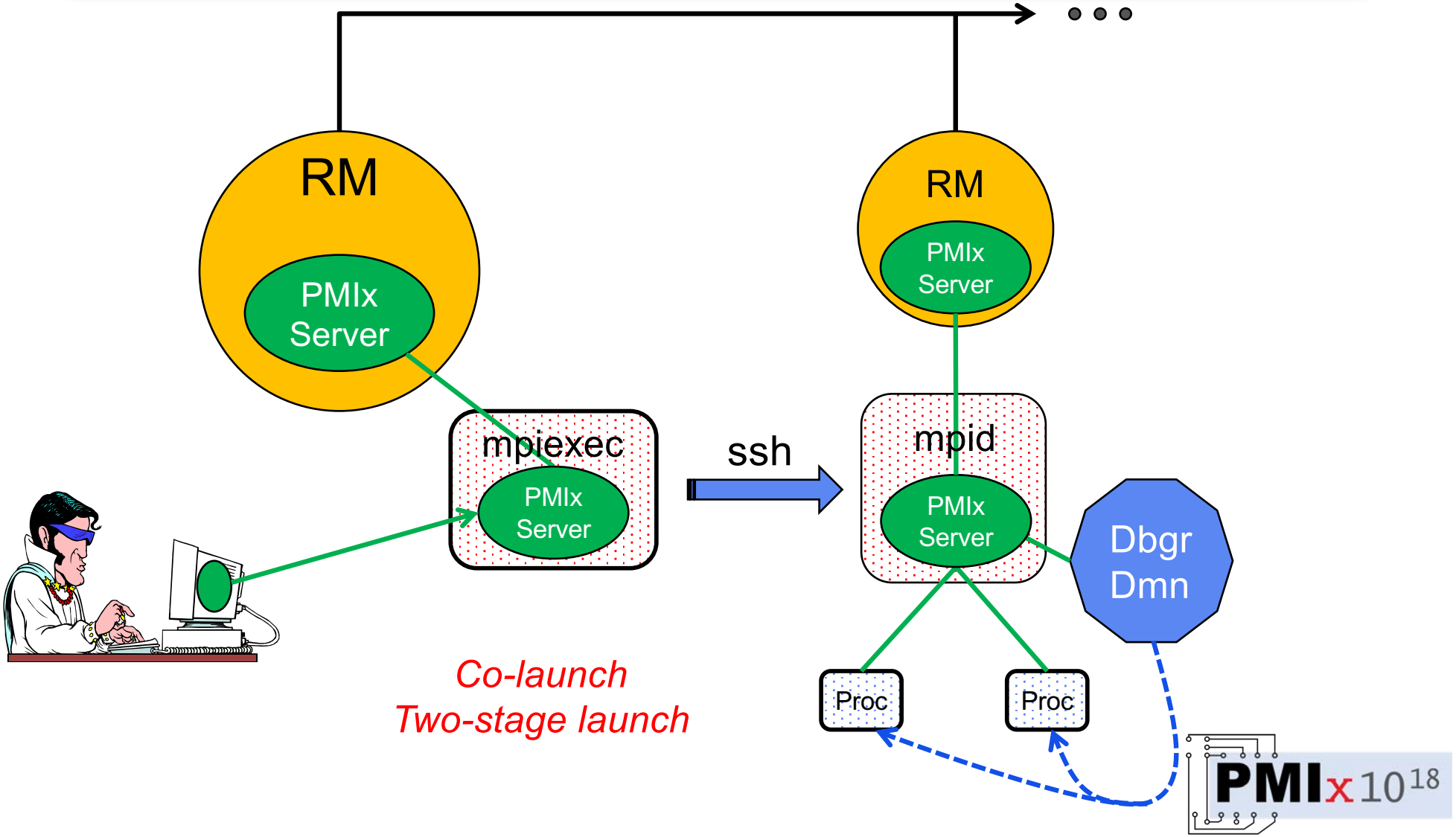


Indirect Launch

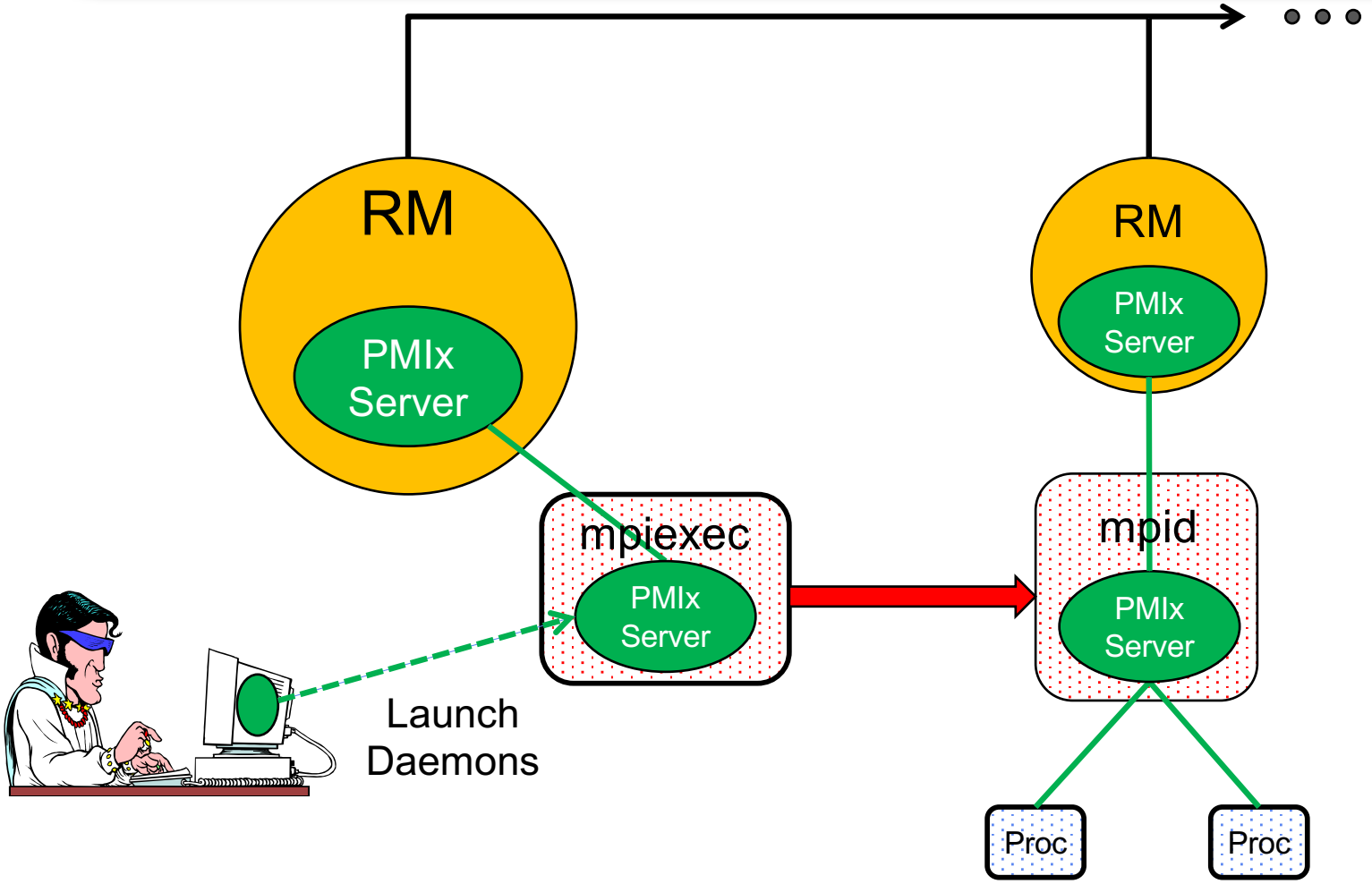


Tools: only connect to one server at a time

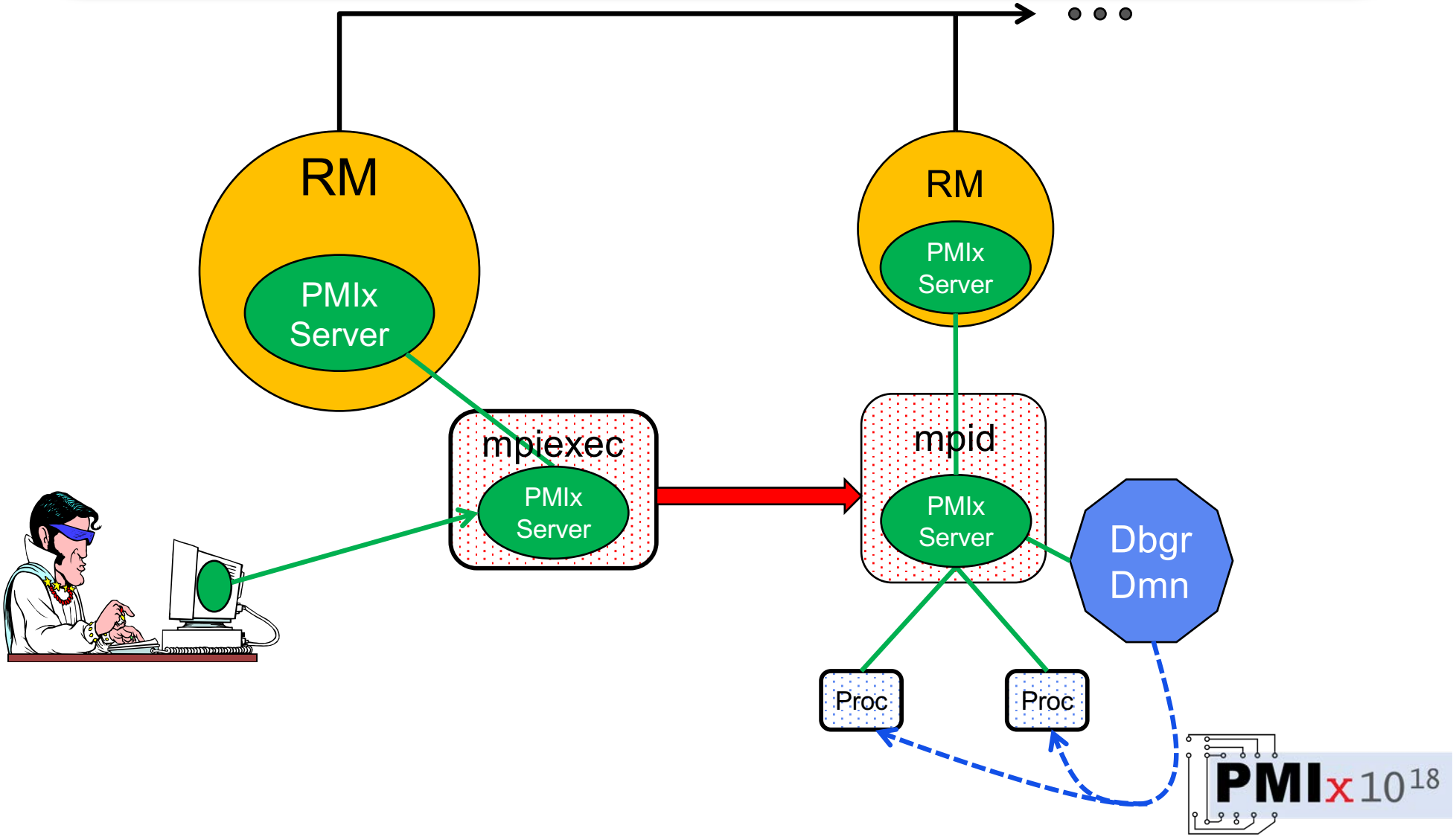
Indirect Launch



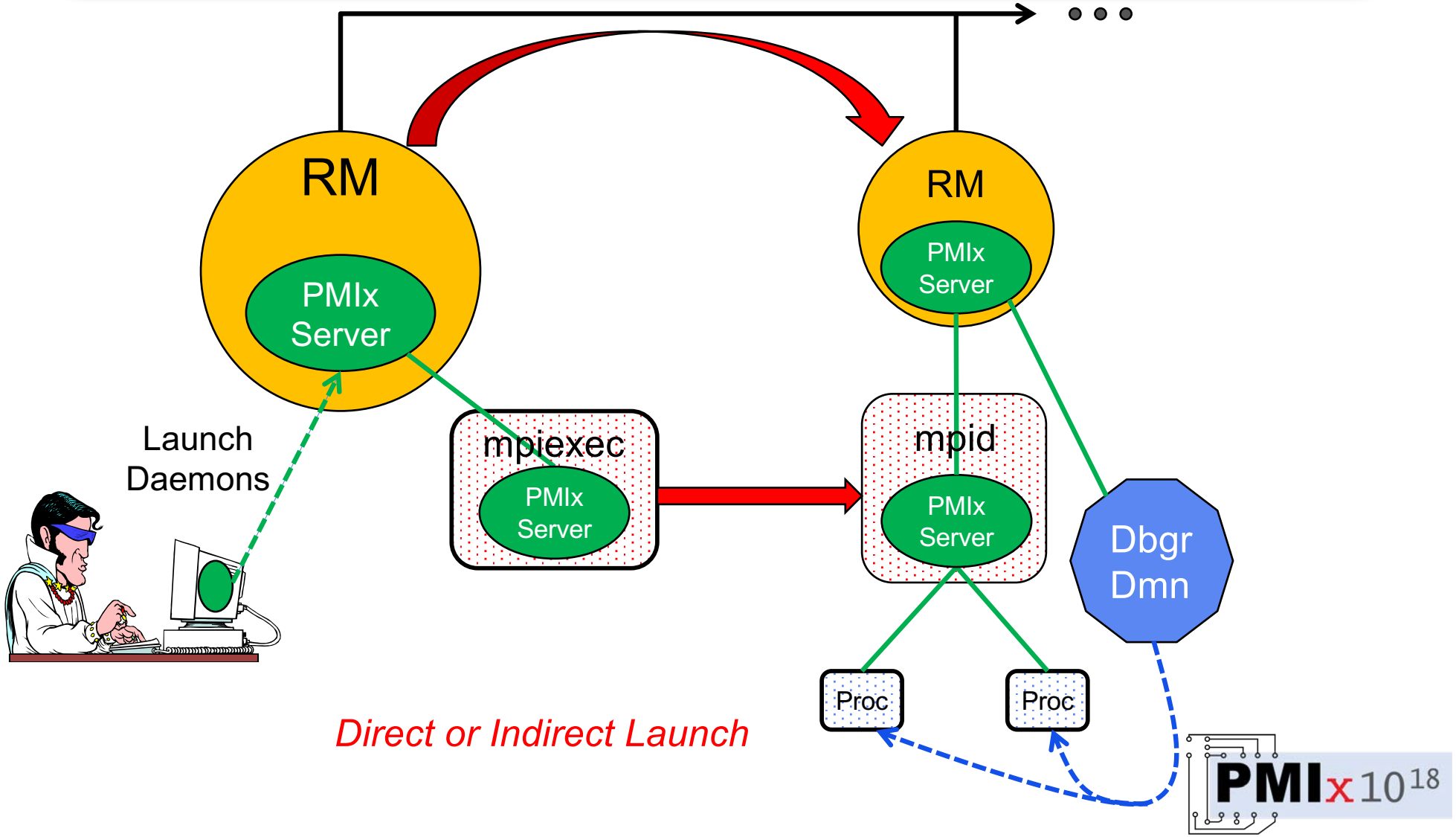
Attach to Running Job



Attach to Running Job



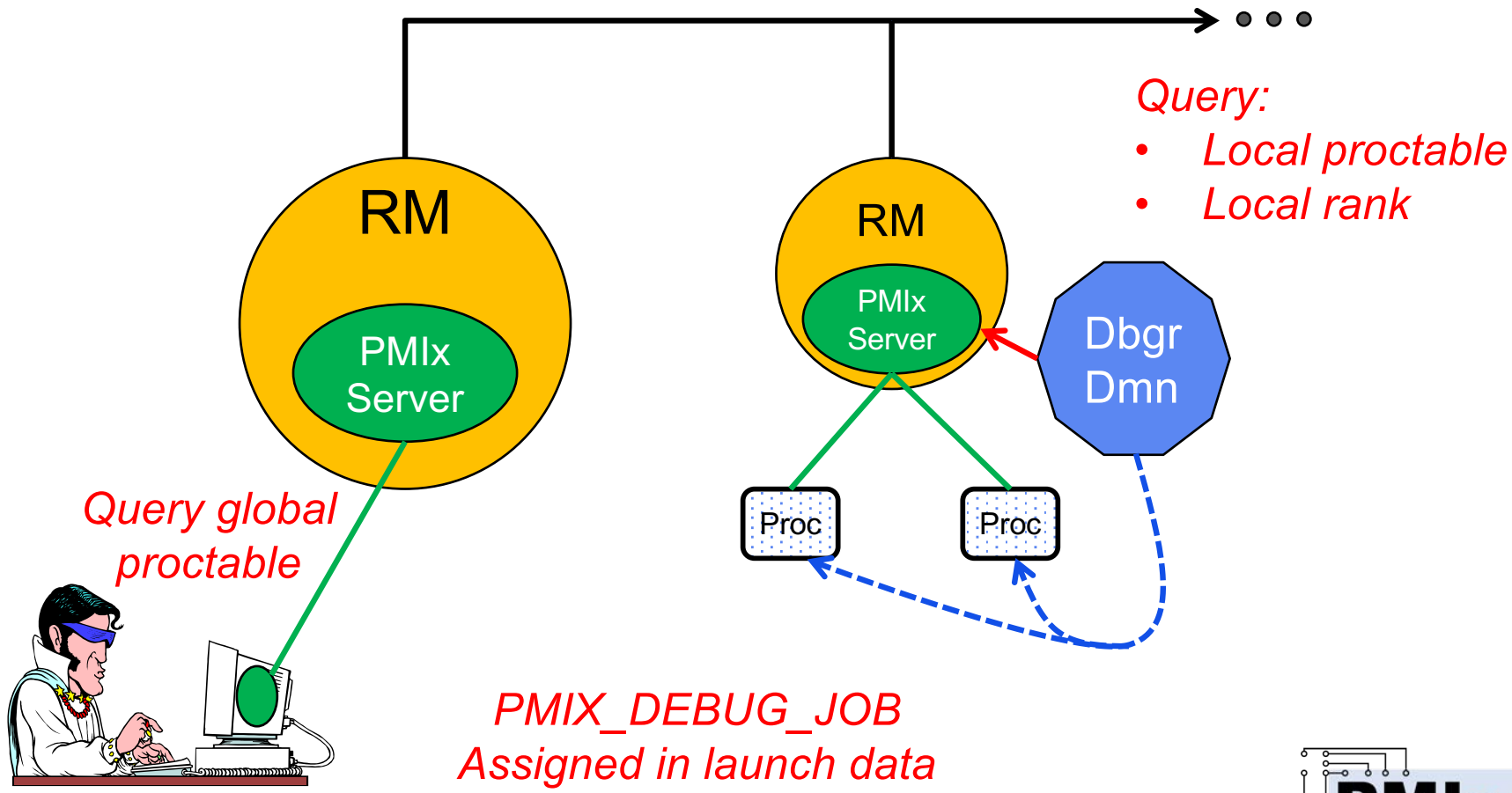
Attach to Running Job



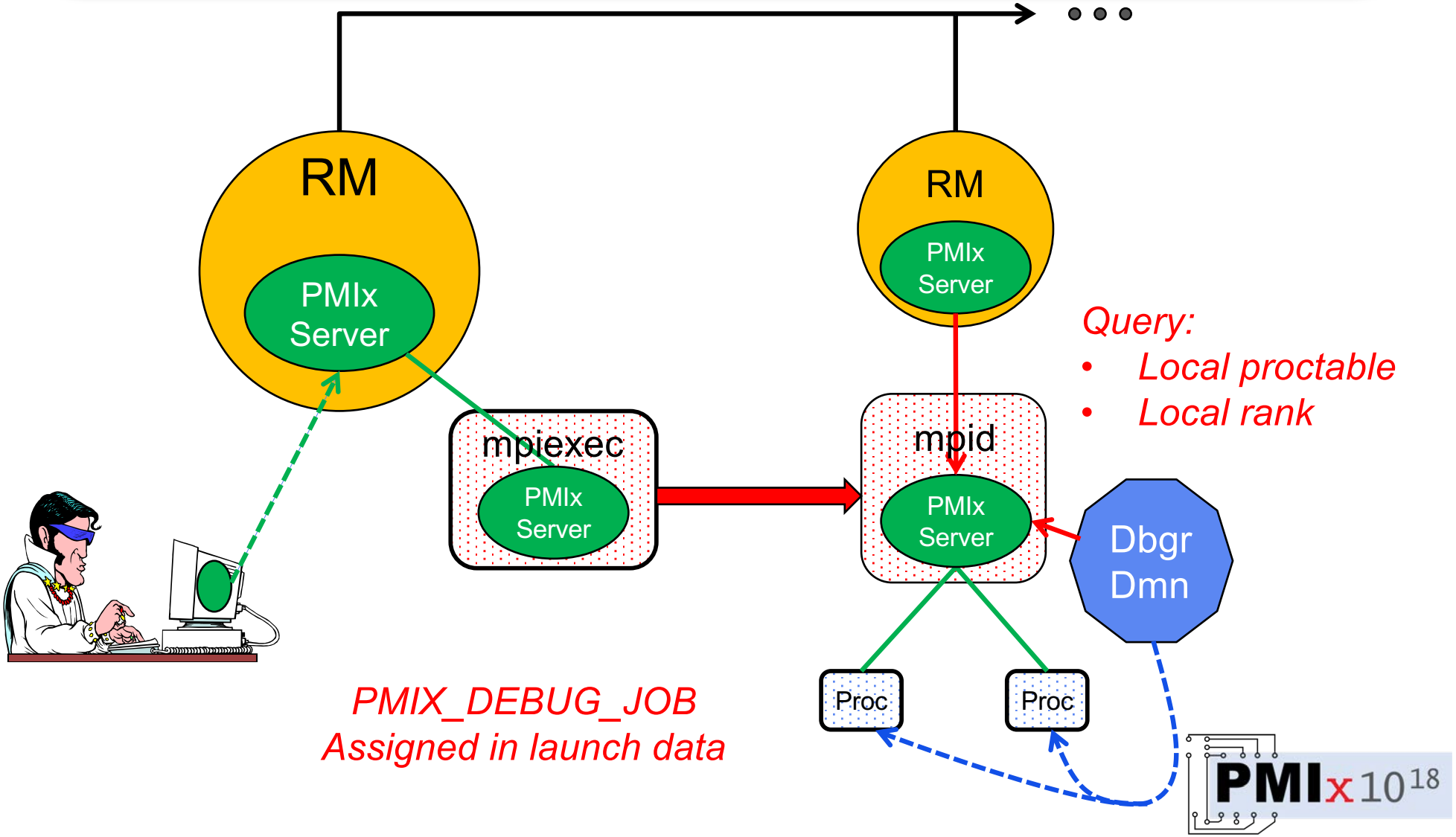
Web Links

- https://pmix.org
  - Home page for PMIx
- https://pmix.org/pmix-standard/
  - Standards process, links to headers, RFCs
  - Roles/responsibilities of RMs and other system management components
- How-to guides
  - https://pmix.org/support/how-to/example-direct-launch-debugger-tool/
  - https://pmix.org/support/how-to/example-indirect-launch-debugger-tool/

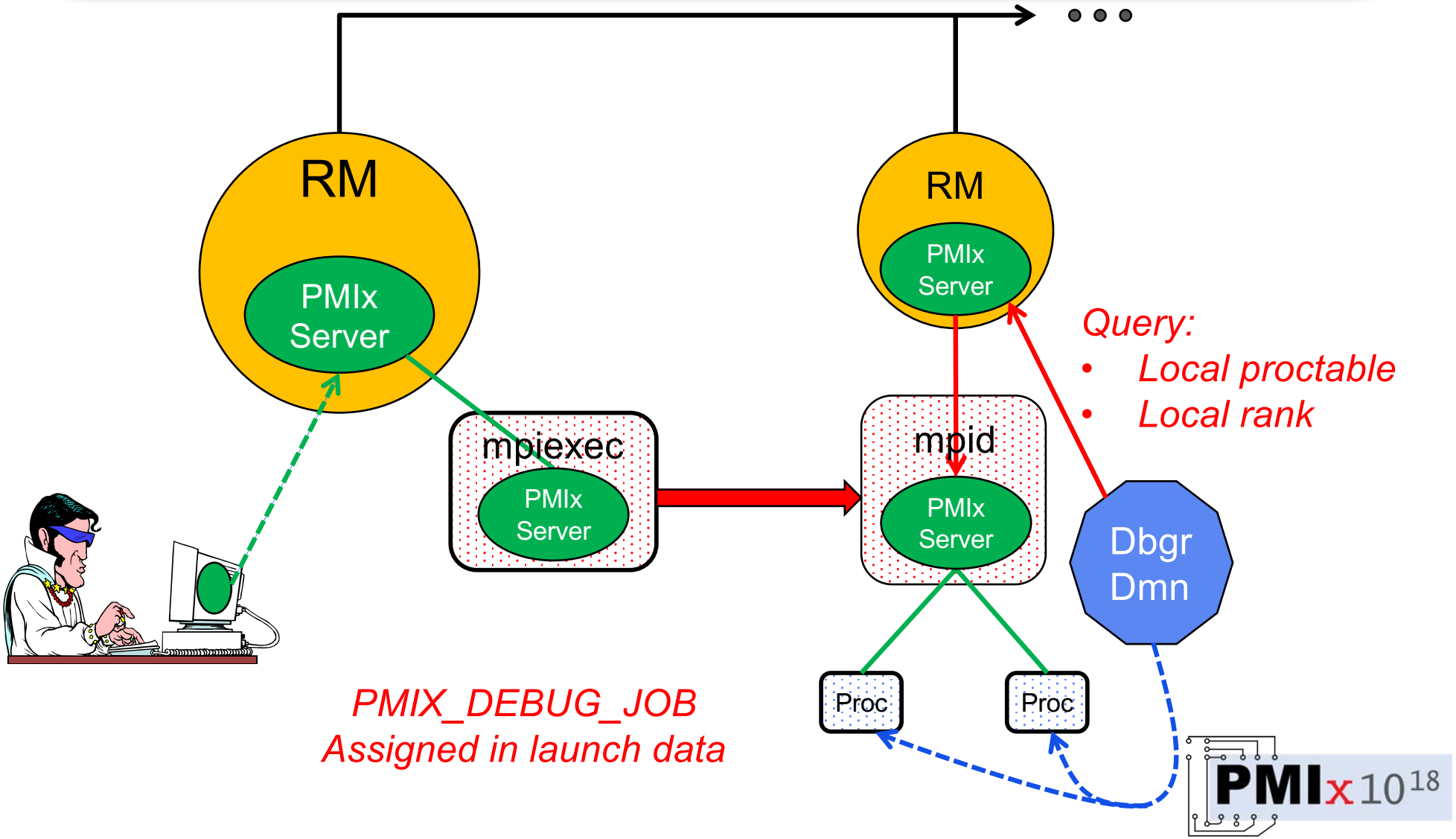
Assigning Procs->Daemons



Assigning Procs->Daemons



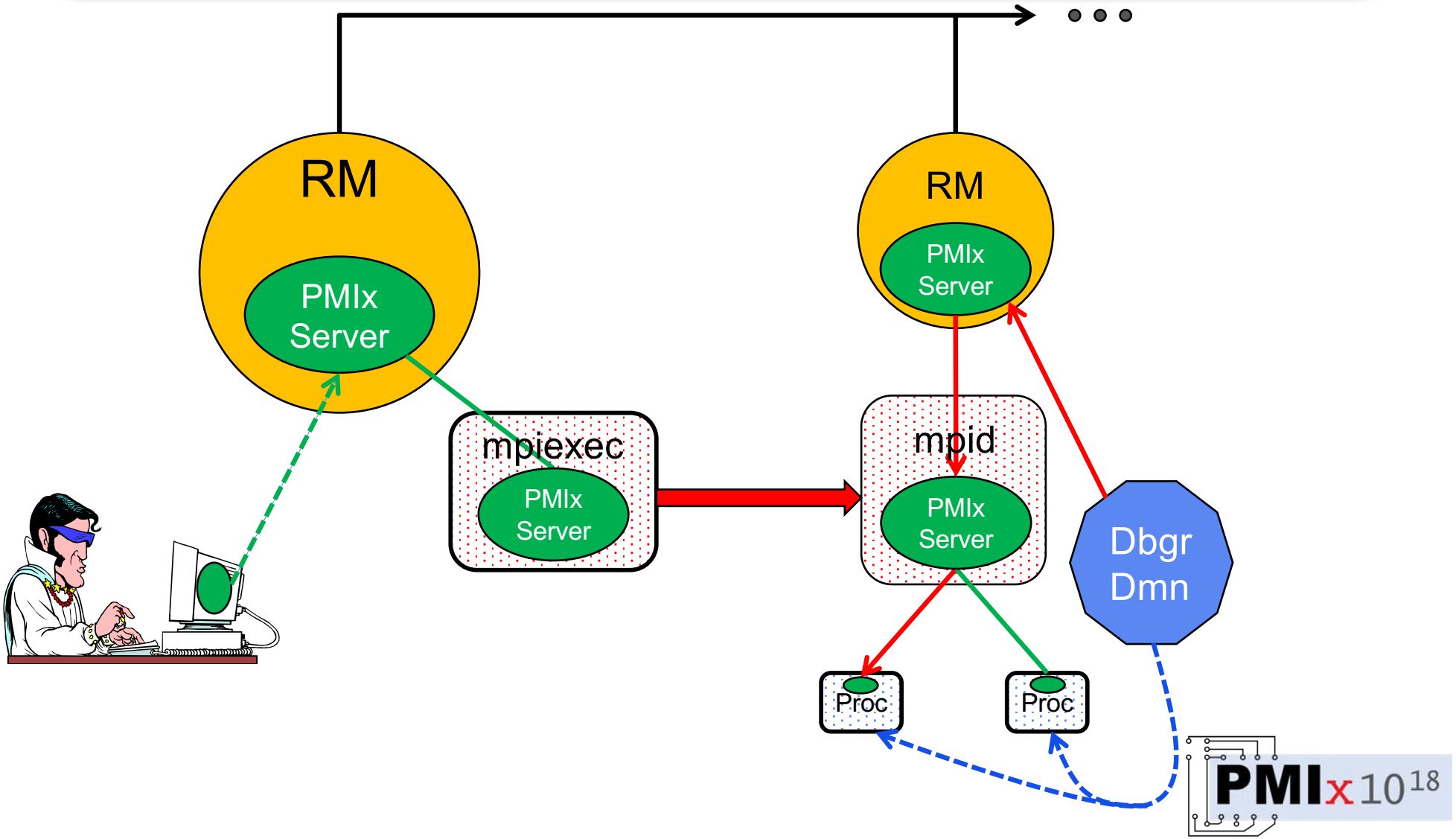
Assigning Procs->Daemons



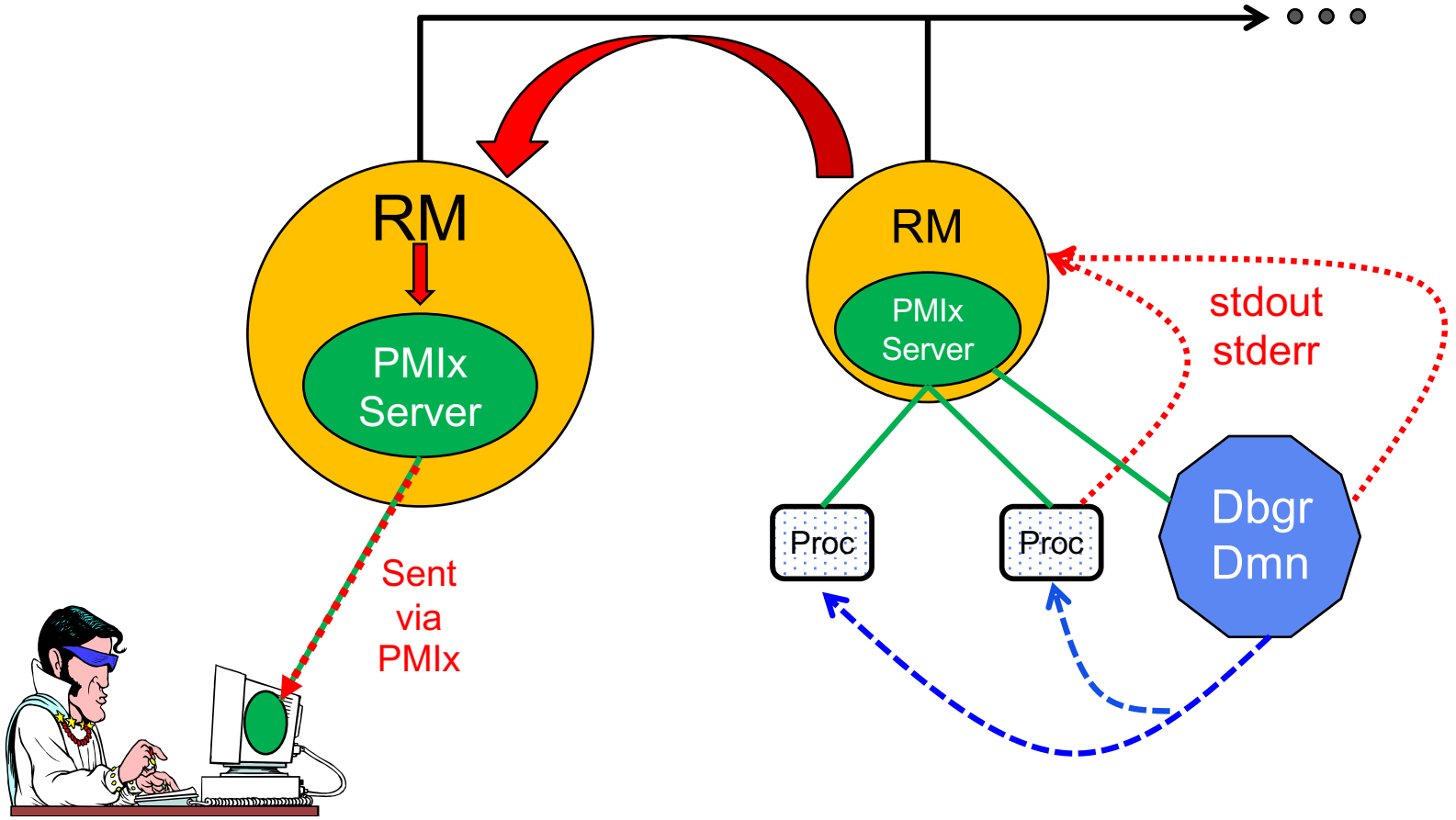
General Capabilities

- Specify mechanism for “hold”
  - Query RM or launcher for support
  - stop-on-exec, stop-in-init, or stop-in-app
- Specify app release mechanism
  - PMIx event, signal, …
- Register for events
  - Termination of debugger job and/or daemons
  - Termination of app job and/or procs

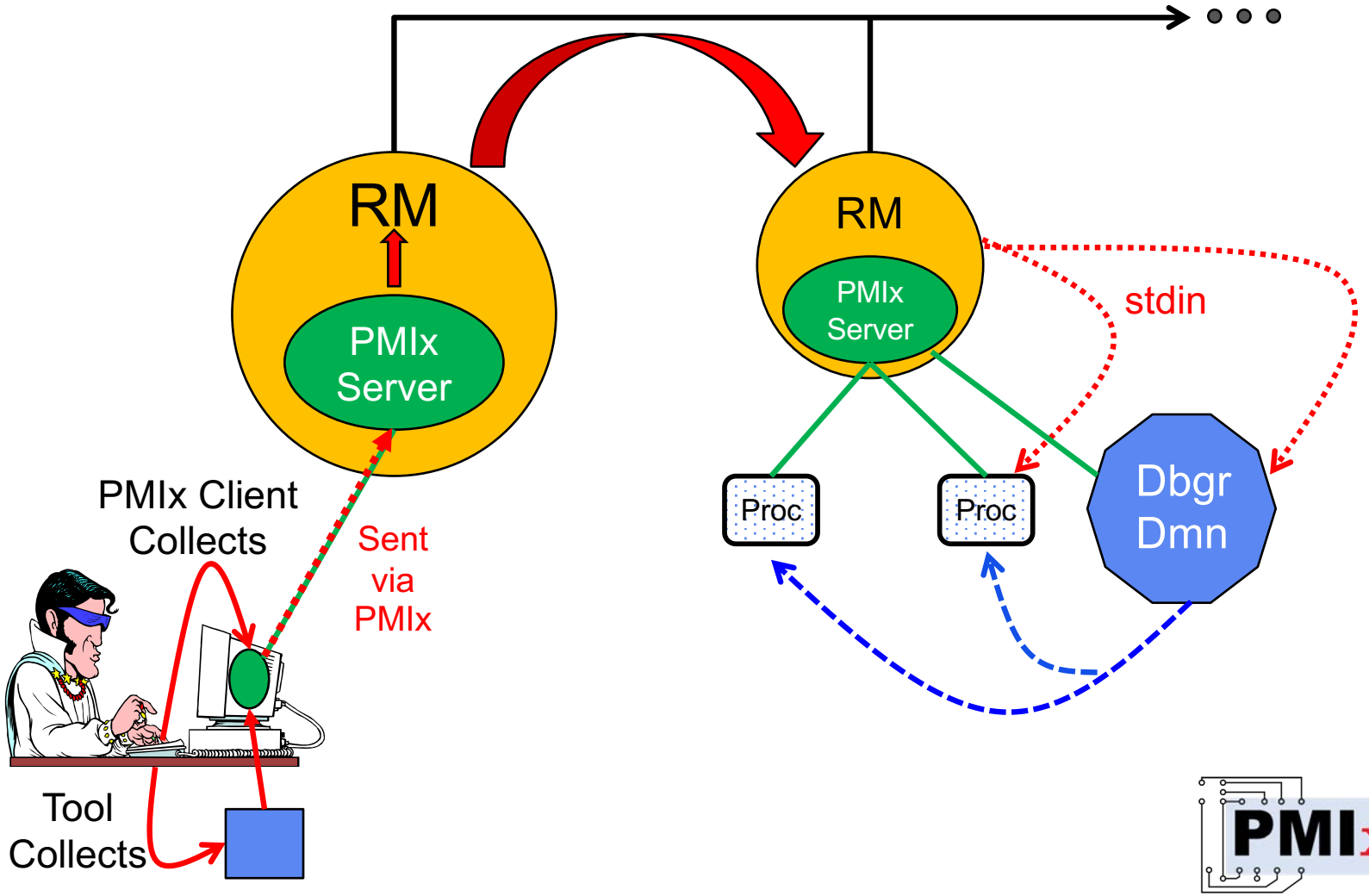
Query App Info



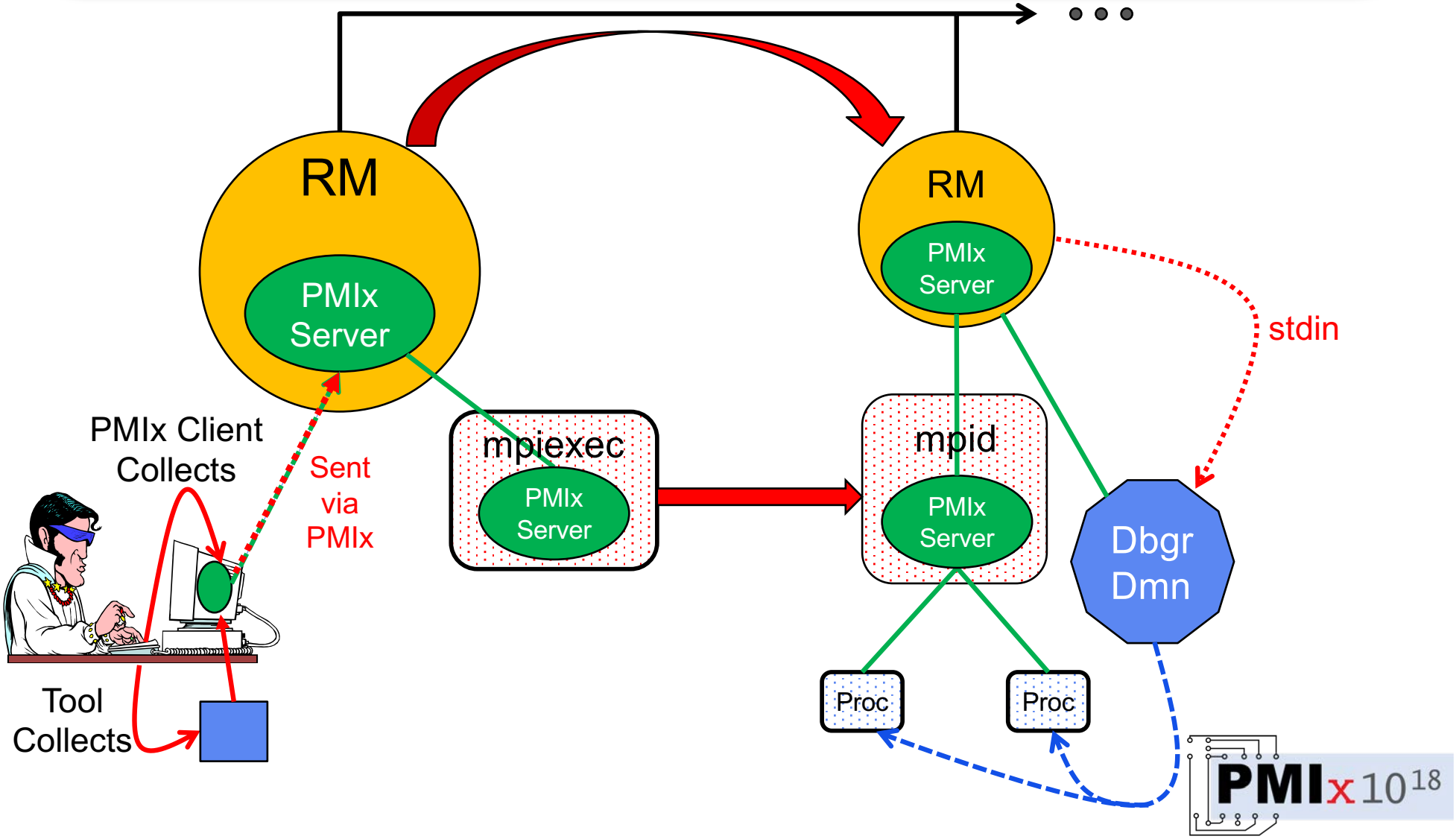
Forwarding of Output



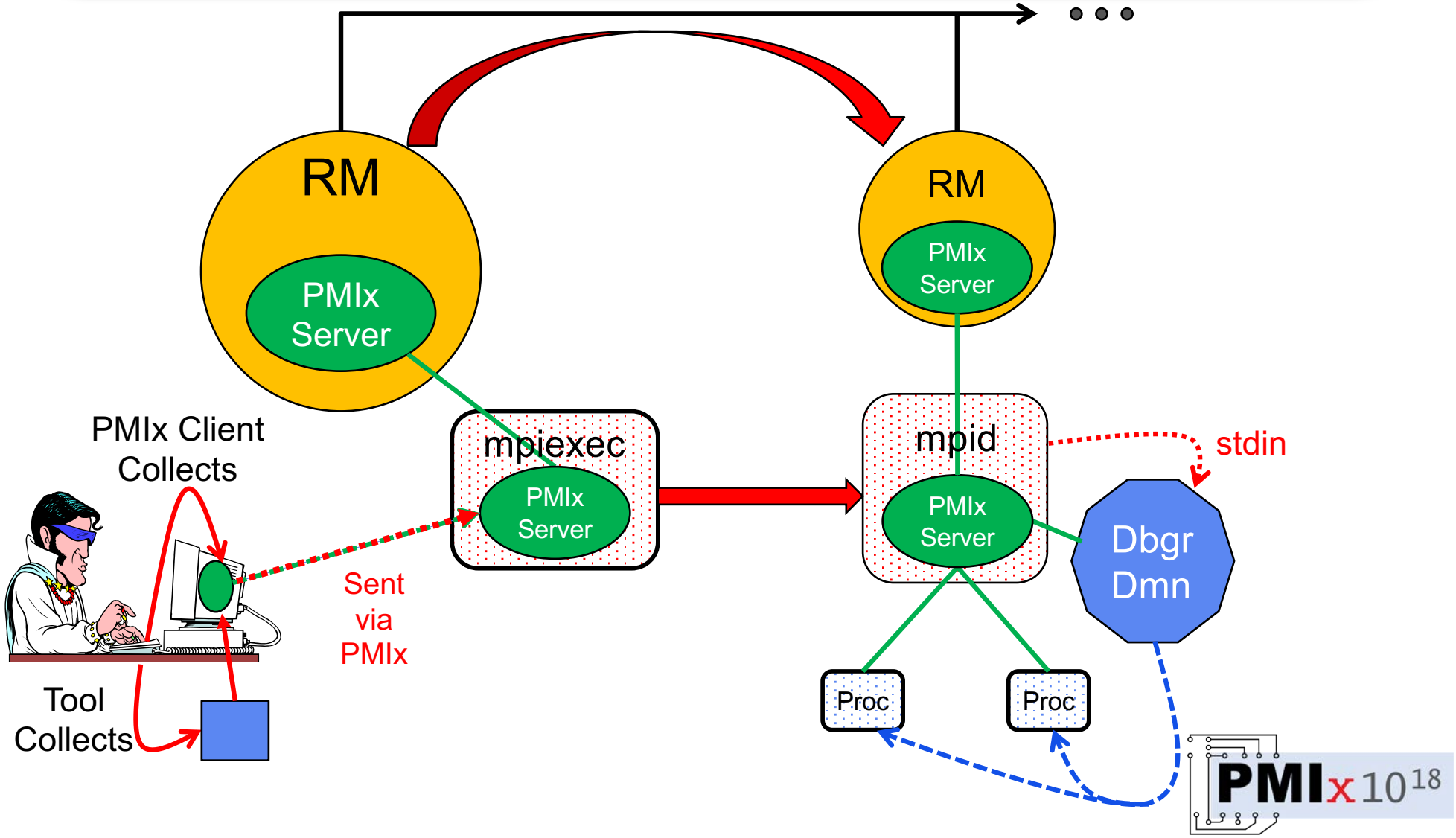
Forwarding Stdin



Forwarding Stdin



Forwarding Stdin



Fabric Features

- Harvest/forward envars
  - Include/exclude
- Get/set security keys
- Get/forward setup “blob”
- Deliver setup “blob” prior to local process spawn

Support:

- *OmniPath (master)*
- *TCP (branch)*