Interoperability and Migration from v3 to v4

Andrew Johnson,
Argonne National Laboratory

Outline

- Terminology
- Architecture
- Interoperability
- Migration
- Impediments
- Conclusions

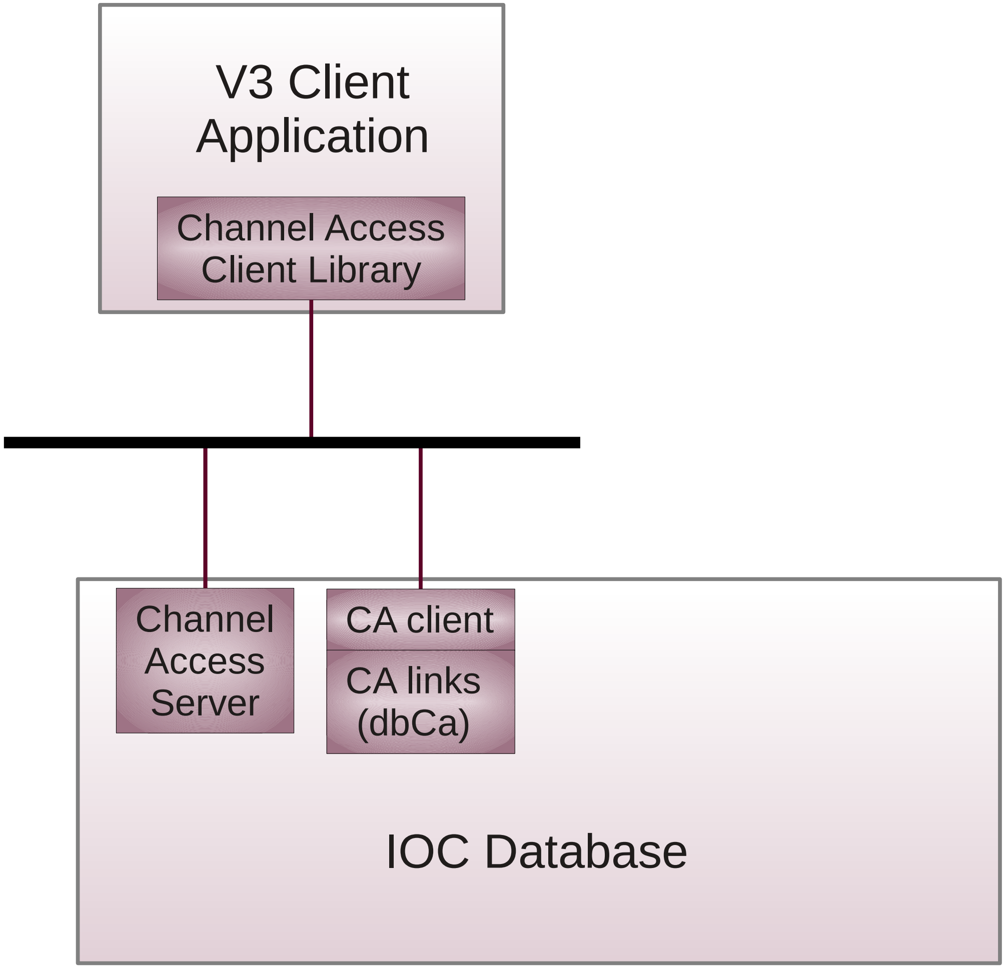
Terminology

- Channel Access (CA)
  - The same network protocol used in 3.14.x, 3.13.x, 3.12.x and earlier versions
    - There are no v4 changes to the CA protocol or the current CA client API
- IOC, IOC Core
  - No change, v4 IOCs use the same database, record and device support as before

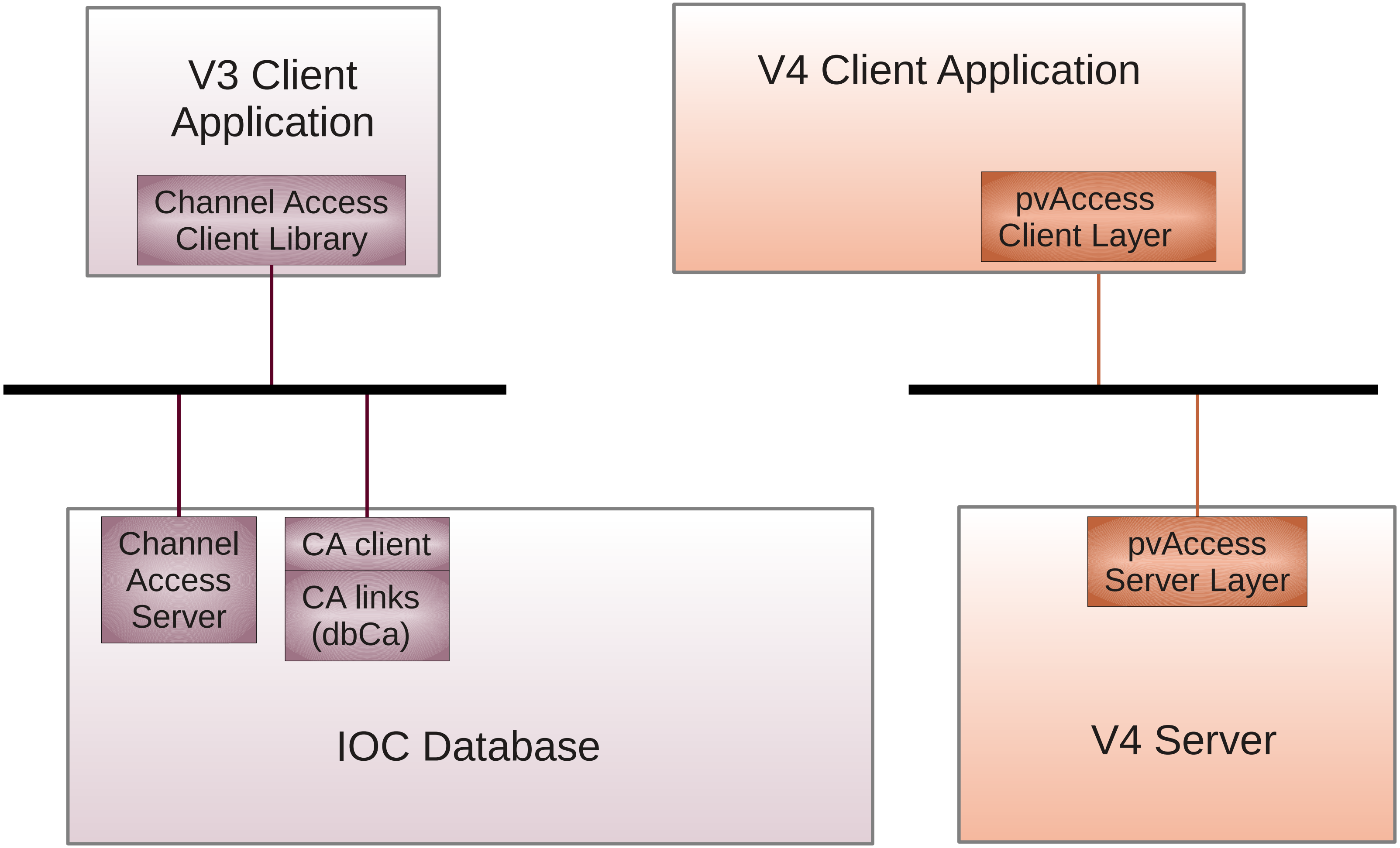
- pvAccess (PVA)
  - New network protocol and its associated APIs, implemented in both C++ and Java
- pvaSrv
  - A pvAccess server for IOC Core, makes fields accessible over PVA network protocol
  - An IOC running pvaSrv may be called a V4 IOC

- Bilingual
  - A client or server that can communicate using CA, PVA or both

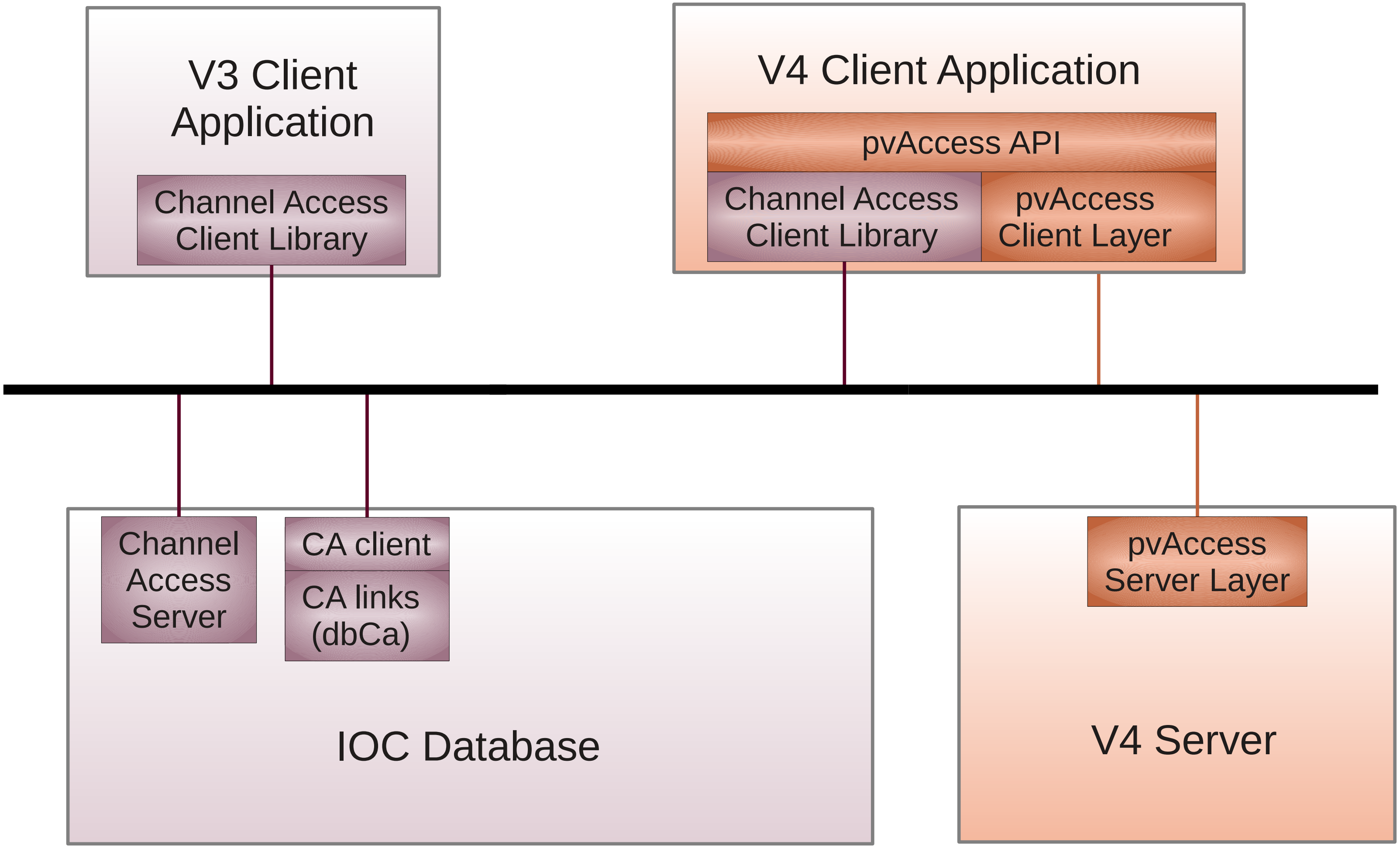
Desired Architecture



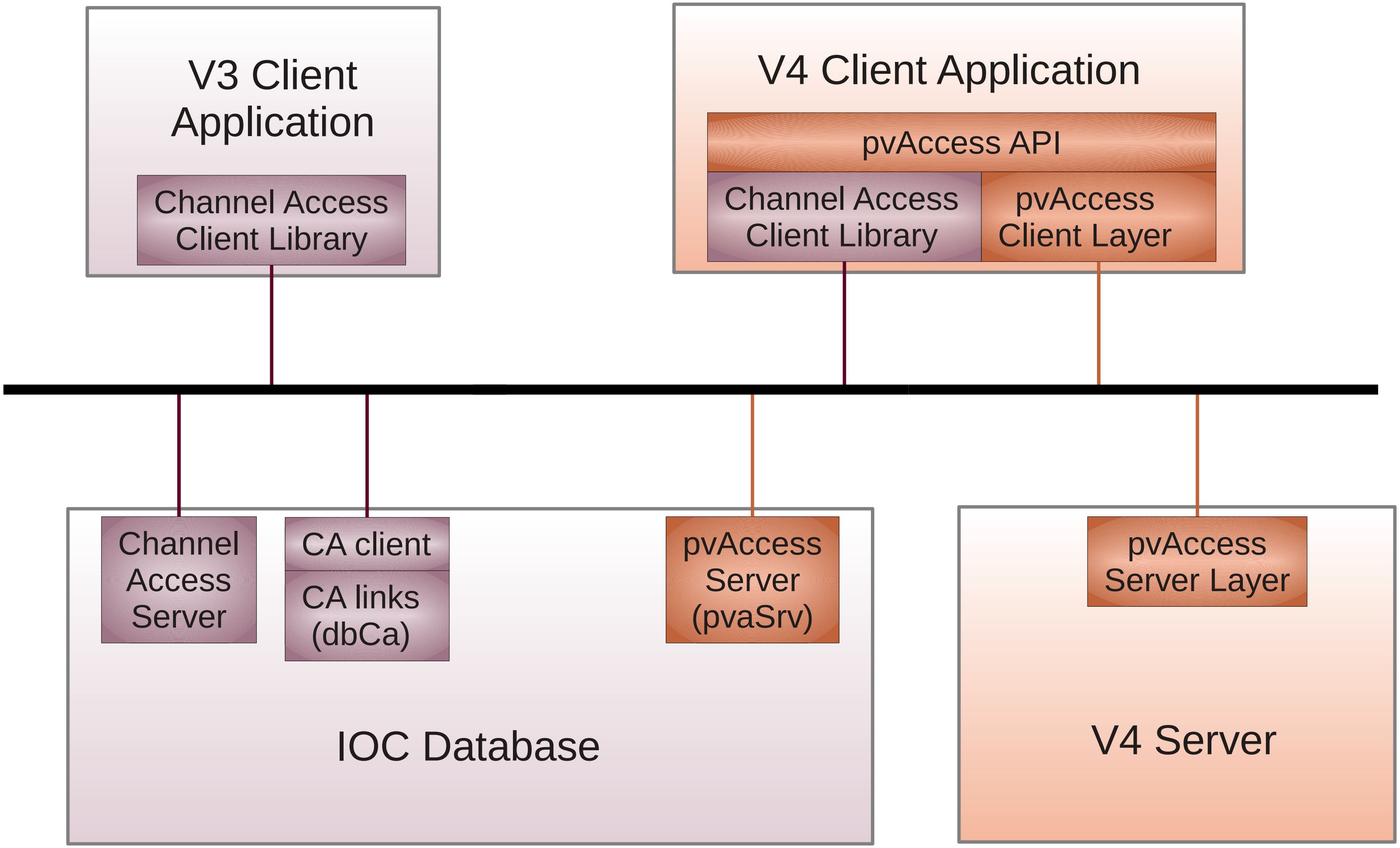
Desired Architecture



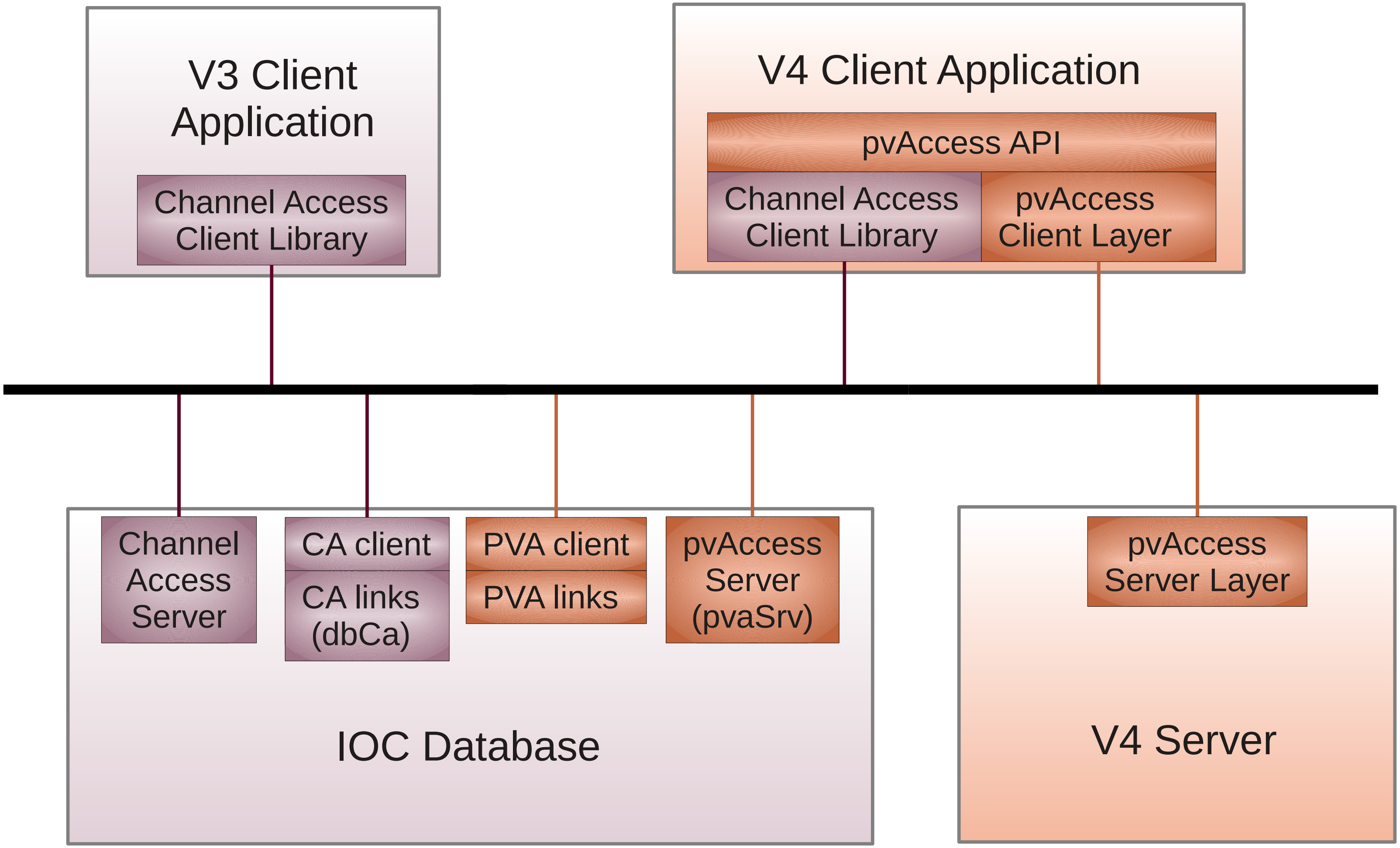
Desired Architecture



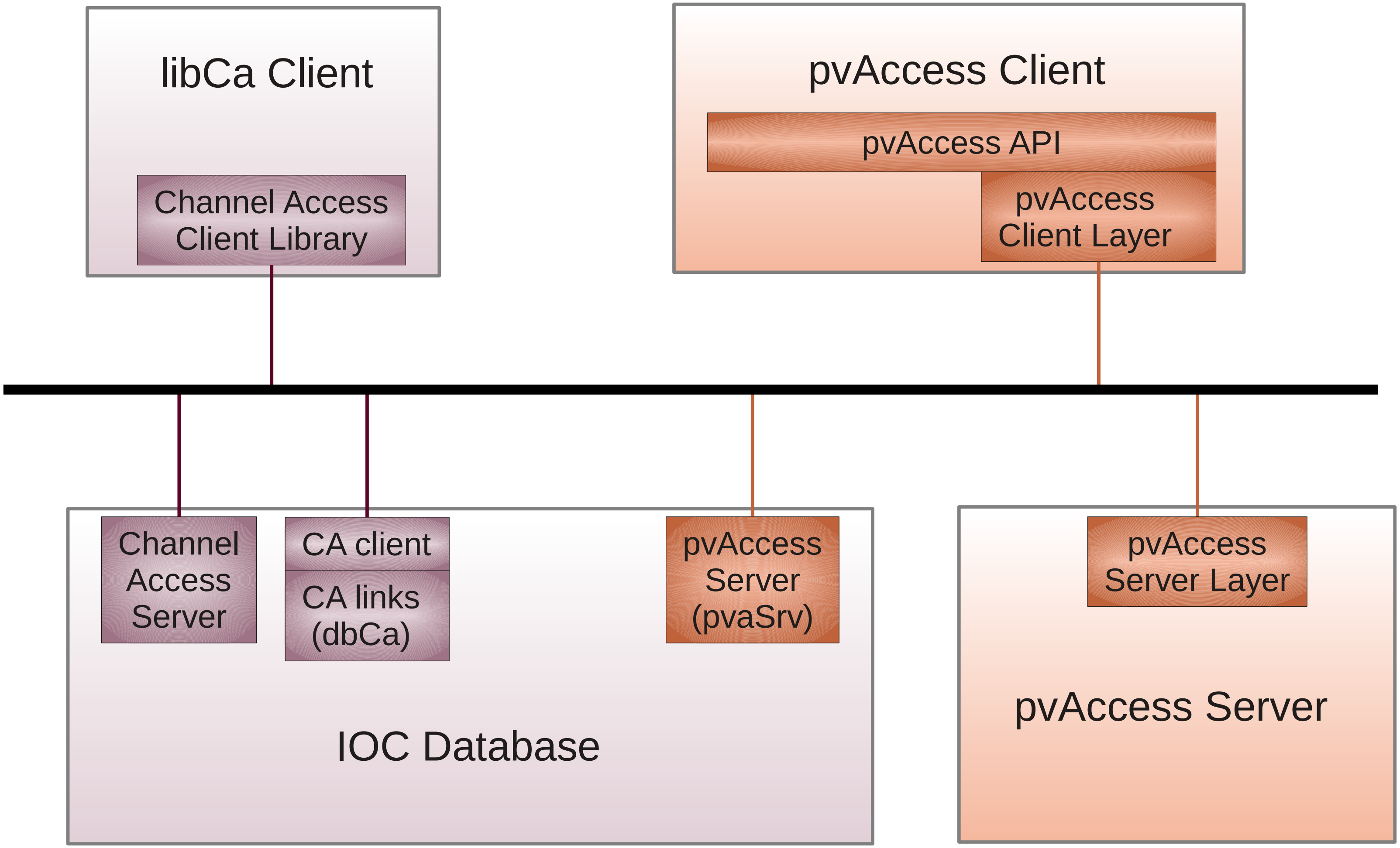
Desired Architecture



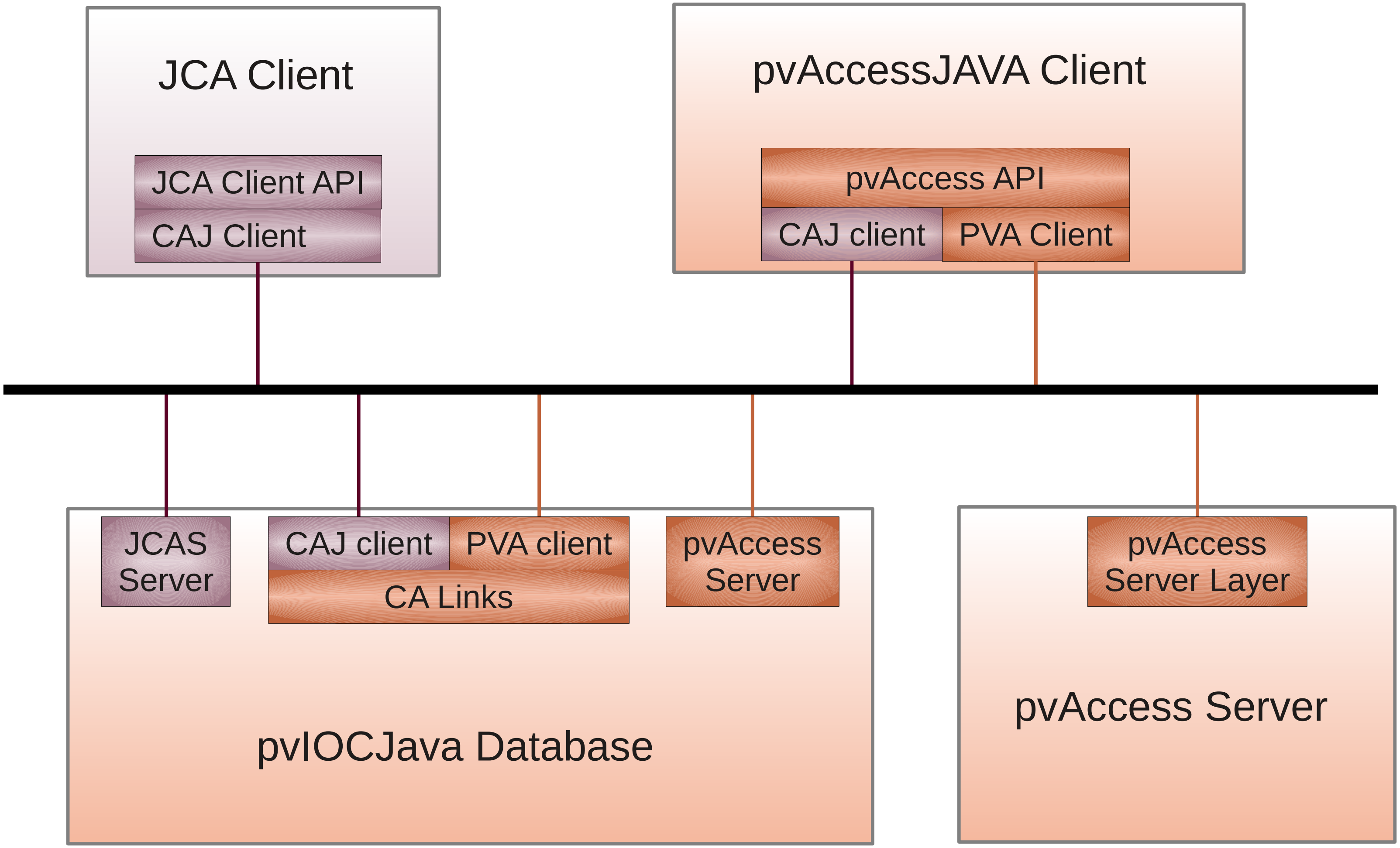
Desired Architecture



C++ Current Implementation



Pure Java Current Implementation



Desired functionality

- Client applications should be able to
  - Exchange complex structured data (pvData) with pvAccess servers (PVA protocol)
  - Communicate with v3 IOCs and other CA servers using just the pvAccess API (CA protocol)
    - Existing client programs must be modified or extended to use the pvAccess API

- IOCs should be able to
  - Publish individual and sets of record fields using a pvAccess server (pvaSrv)
    - Atomic access to multiple records or multiple fields on one IOC as a structure or array
  - Point record link fields to pvAccess channels as well as CA channels
    - Full bilingual link support, not just soft device support

Interoperability

- Bilingual Clients
  - pvAccess clients written in Java can be made bilingual
    - The application just calls a different channel provider to connect over CA
    - The CA channel provider is not available for C++ clients yet
  - Bilingual clients can communicate with existing IOCs without them having to run pvaSrv

- Bilingual IOCs
  - Any IOC with pvaSrv running is a bilingual server
    - Record fields can be accessed over either CA or PVA
    - Atomic multi-record or multi-field access is not supported yet
  - Writing device support that addresses pvAccess channels should be relatively simple
  - Fully supporting PVA links in all record link fields will be a major project
    - Resurrecting parts of the Link Support design I presented in 1999
    - This might break backwards compatibility for some PV names

Migration

- How to get from v3 to v4?
  - Control System Studio users
    - Upgrade to a version of CSS than includes pvAccess support
  - Python developers
    - Work has started on a Python API for pvAccess — help welcome
  - IOC developers can add pvaSrv to existing 3.14.12 IOCs
    - Add a few v4 modules to the RELEASE file
    - Add one DBD file and some libraries to the IOC in its Makefile
    - Add two lines to the IOC start-up script
- IOCs will have to run EPICS 3.15 to use future versions of pvAccess
  - Modifications to IOC Core will be needed for some features

Potential Impediments

- Access Security is not yet implemented in pvAccess
- No PV Gateway for pvAccess protocol
  - Sites with firewall-protected IP sub-nets can’t migrate yet
  - Multi-cast IP and the Channel Finder *can* be used to cross sub-nets
- IOCs running older versions of Base or VxWorks must be upgraded
  - Base 3.14.12.3 (will soon change to 3.15.x)
  - VxWorks 5.5 or later
- Microsoft Windows not yet supported

Conclusion

- EPICS v4 can be introduced into existing v3 control systems
  - pvAccess clients written in Java can talk to existing unmodified IOCs
    - Bilingual versions of CSS can now talk both pvAccess and Channel Access protocols
  - Adding a pvAccess server to an EPICS 3.14.12 IOC is simple

- Future versions of pvAccess will provide additional features
  - Sites with fire-walled IP subnets that rely on the PV Gateway could help develop one for pvAccess