Development Team Scenarios

Overview

QlikView is an extremely flexible and easily adapted BI tool.   As such, development teams can organize around several models for support, administration, development, training and management.   These scenarios help guide discussions about possible configurations for QlikView roles in a development environment.

It is recommended that the client consult its own IT standards for development, as they may drive this decision, or at least narrow the allowed choices.

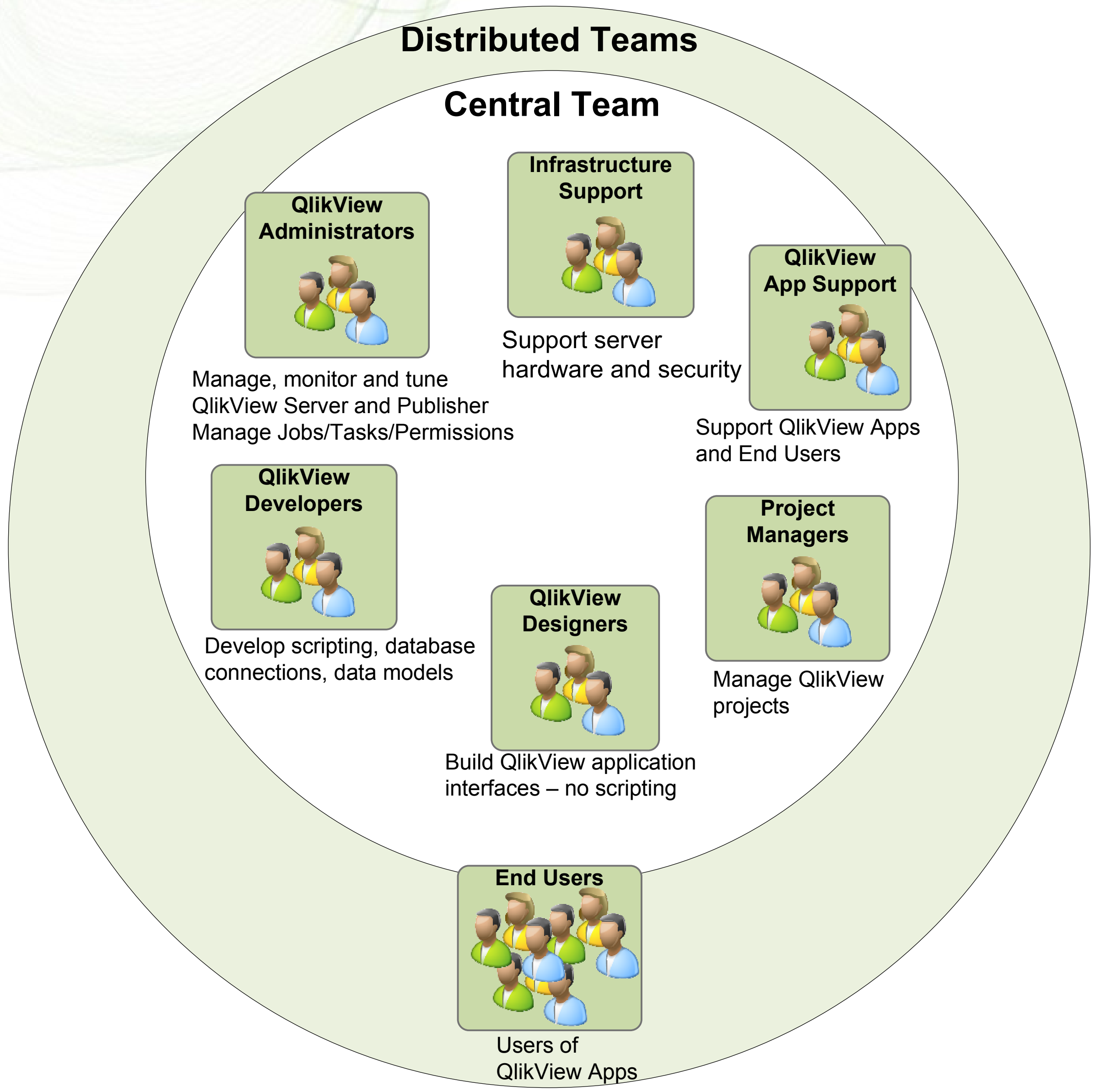
QlikTech does not expressly promote one of these scenarios over the others, but asks that clients determine for themselves which of these configurations might work best, given the nature of the QlikView development and the skills sets that exist. Use these scenarios as templates to start your own diagram that is tailored to your needs.

The subsequent pages in this document describe the scenarios.

Author: BPN, QlikTech NA

Development Teams – Scenario #1 (Fully Centralized)

Scenario Description: This scenario depicts a full centralized development team. Departments don't need to supply developers, support personnel or administrators to use QlikView applications. They request new applications and then consume them along with central QlikView services.



Distributed Teams
Central Team
QlikView Administrators
Manage, monitor and tune QlikView Server and Publisher Manage Jobs/Tasks/Permissions
Infrastructure Support
Support server hardware and security
QlikView App Support
Support QlikView Apps and End Users
QlikView Developers
Develop scripting, database connections, data models
Project Managers
Manage QlikView projects
QlikView Designers
Build QlikView application interfaces – no scripting
End Users
Users of QlikView Apps

Strengths:

- skill set sharing
- data mgmt and re-use
- data access control
- increased consistency
- best practices are easier to govern
- lower TCO over time.

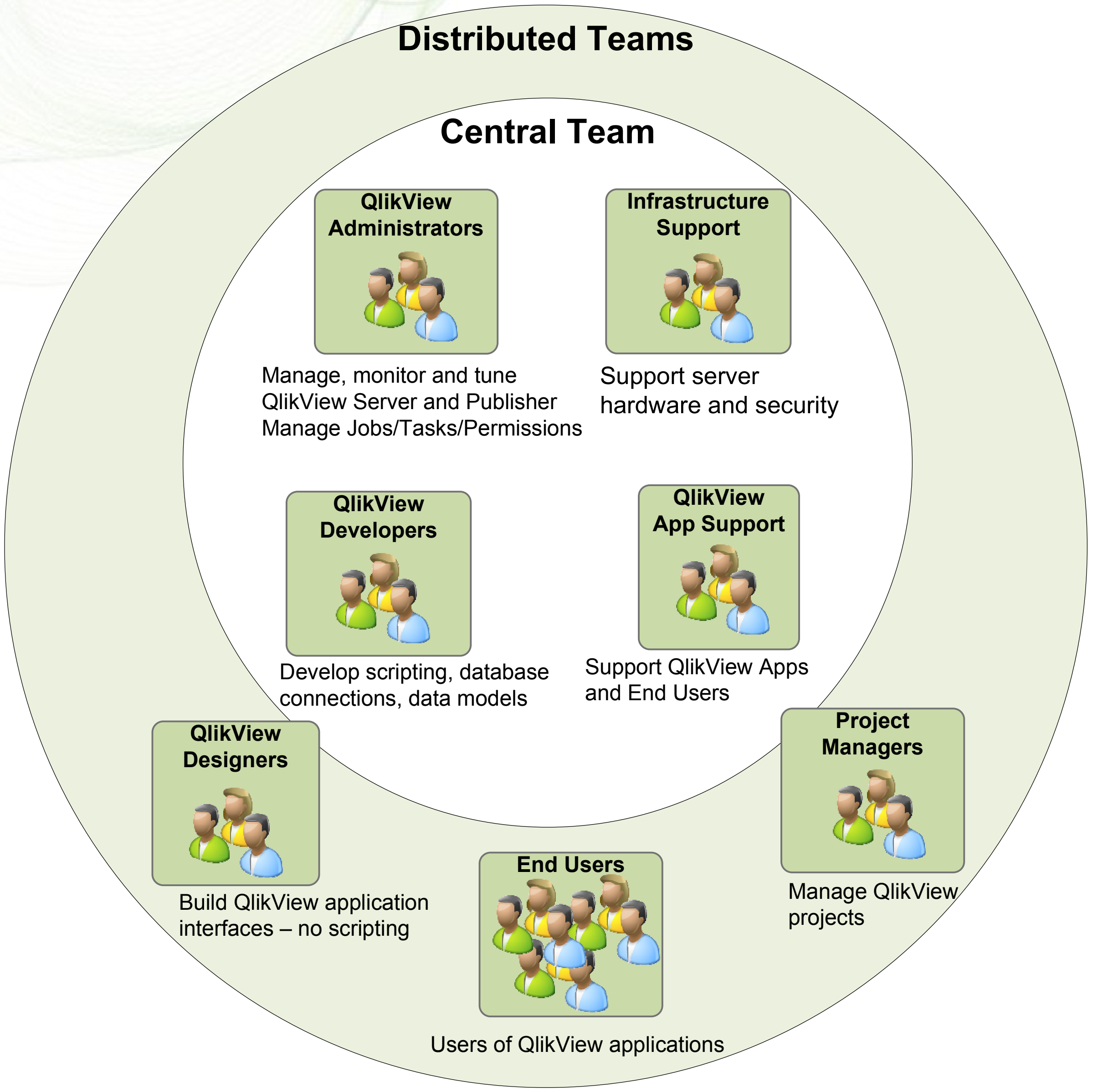
Challenges:

- lack of subject matter expertise in departmental subjects
- possible bottlenecks for services and development
- need chargeback or cost tracking to allocate costs to departments

Author: BPN, QlikTech NA

Development Teams – Scenario #2 (Mostly Centralized)

Scenario Description: This scenario depicts a mostly centralized development team. Enterprise development is retained as a central function, allowing for the scripting and data modeling to be handled by expert QlikView developers and data professionals. Departments are responsible for all training, project mgmt, application design, testing and support.


Distributed Teams
Central Team
QlikView Administrators
Manage, monitor and tune QlikView Server and Publisher Manage Jobs/Tasks/Permissions
Infrastructure Support
Support server hardware and security
QlikView Developers
Develop scripting, database connections, data models
QlikView App Support
Support QlikView Apps and End Users
QlikView Designers
Build QlikView application interfaces – no scripting
Project Managers
Manage QlikView projects
End Users
Users of QlikView applications

Strengths:

- skill set sharing
- data mgmt and re-use
- data access control
- Project Mgmt and Design still located in departments where subject matter experts reside

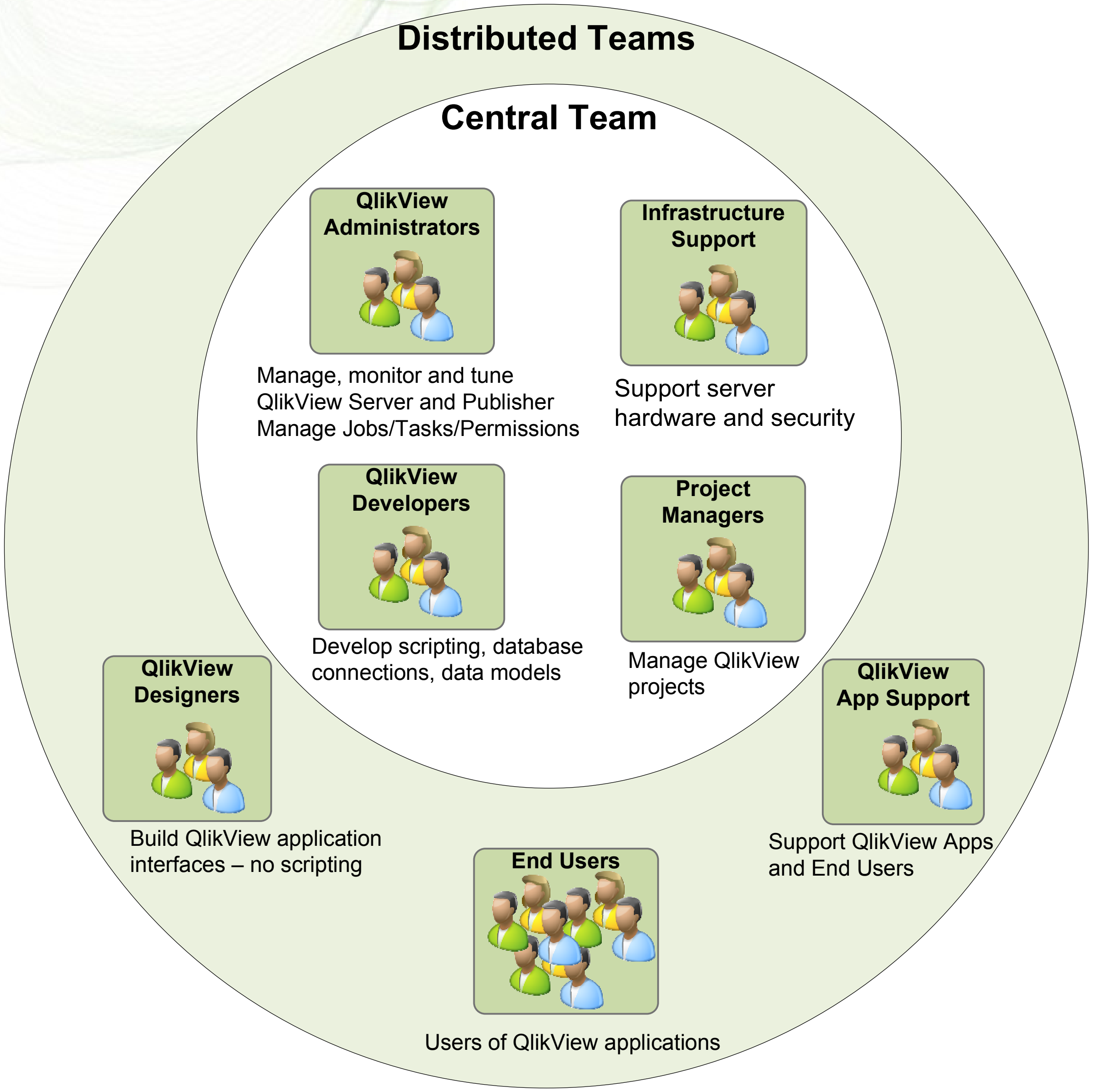
Challenges:

- lack of design consistency
- processes governance for departmental Professional Developers
- resource bottlenecks for central team resources
- central support may lack subject matter expertise

Author: BPN, QlikTech NA

Development Teams – Scenario #3 (Mostly Centralized)

Scenario Description: This scenario depicts another mostly centralized development team. Support has been moved to departments, but Project Mgmt is retained in the central team to better allow for QlikView expertise and control of designs. Departments are responsible for all training, application design, and support.


Distributed Teams
Central Team
QlikView Administrators
Manage, monitor and tune QlikView Server and Publisher Manage Jobs/Tasks/Permissions
Infrastructure Support
Support server hardware and security
QlikView Developers
Develop scripting, database connections, data models
Project Managers
Manage QlikView projects
QlikView Designers
Build QlikView application interfaces – no scripting
QlikView App Support
Support QlikView Apps and End Users
End Users
Users of QlikView applications

Strengths:

- skill set sharing
- data mgmt and re-use
- data access control
- Support still located in departments where subject matter experts reside

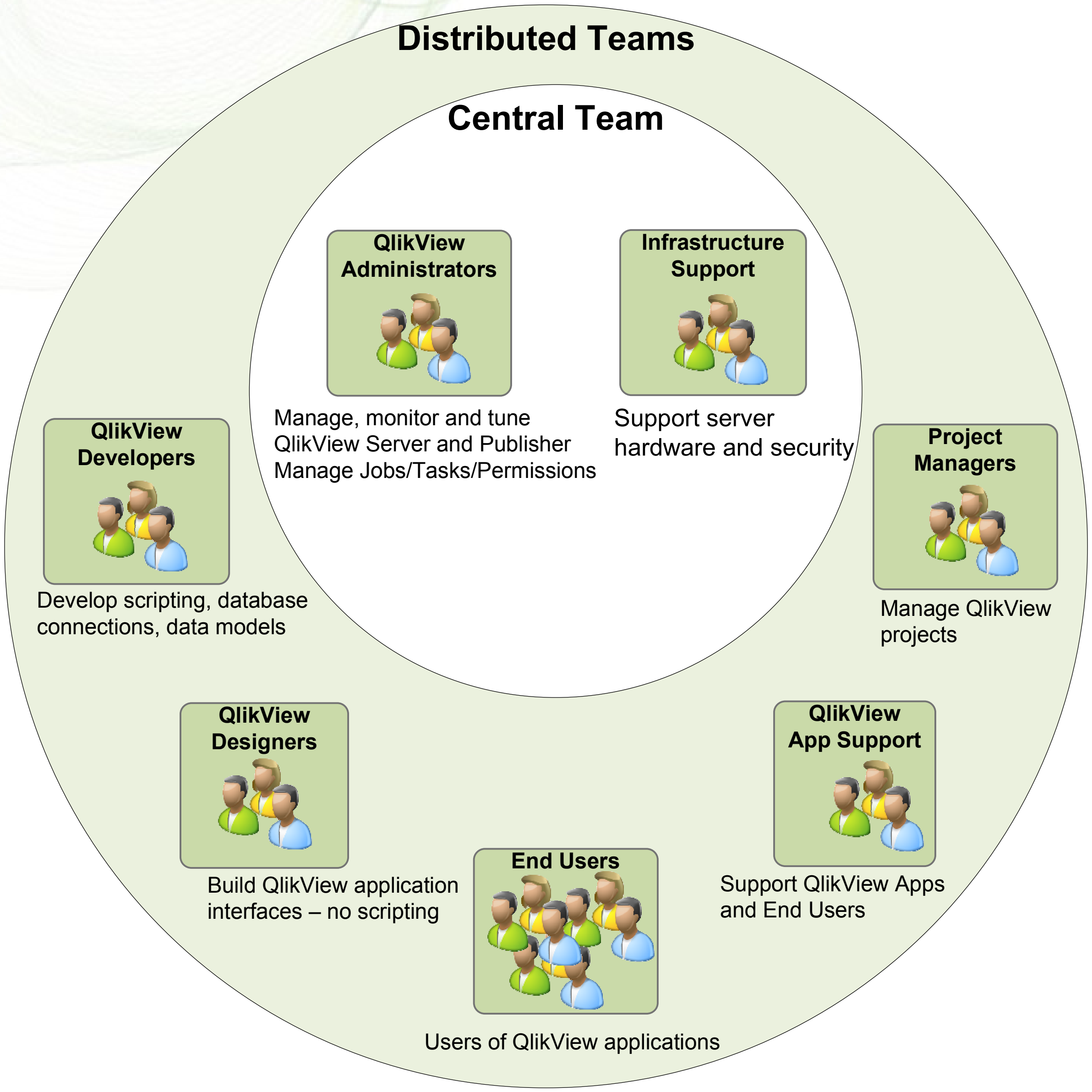
Challenges:

- design consistency
- processes governance for departmental Professional Developers
- resource bottlenecks for central team resources

Author: BPN, QlikTech NA

Development Teams – Scenario #4 (Mostly Decentralized)

Scenario Description: This scenario depicts a mostly decentralized development team, but with retention of centralized infrastructure and administration for QlikView. Departments are responsible for all training, project mgmt, application design, scripting, development, testing and support.



Distributed Teams
Central Team
QlikView Administrators
Manage, monitor and tune QlikView Server and Publisher Manage Jobs/Tasks/Permissions
Infrastructure Support
Support server hardware and security
QlikView Developers
Develop scripting, database connections, data models
Project Managers
Manage QlikView projects
QlikView Designers
Build QlikView application interfaces – no scripting
QlikView App Support
Support QlikView Apps and End Users
End Users
Users of QlikView applications

Strengths:

- minimal central team training
- server/publisher consistency
- flexible, fast development
- most costs born by departments

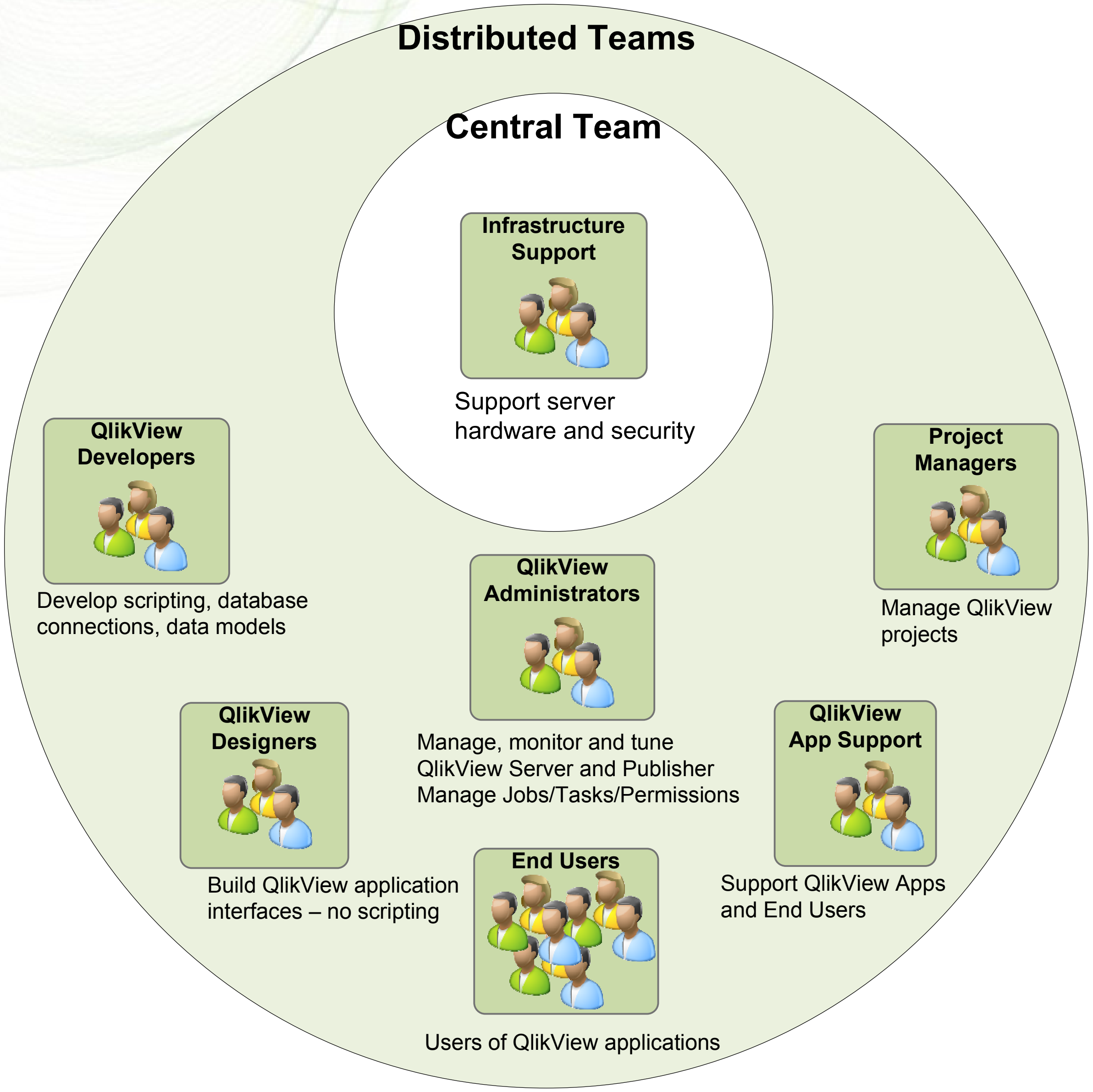
Challenges:

- lack of development consistency
- lack of re-use and standardization
- lack of skill set sharing across departments

Author: BPN, QlikTech NA

Development Teams – Scenario #5 (Fully Decentralized)

Scenario Description: This scenario depicts a fully decentralized development team. Infrastructure is still maintained centrally, but all other aspects of QlikView development, testing, support, training a usage are distributed to departments.


Distributed Teams
Central Team
Infrastructure Support
Support server hardware and security
QlikView Developers
Develop scripting, database connections, data models
Project Managers
Manage QlikView projects
QlikView Administrators
Manage, monitor and tune QlikView Server and Publisher Manage Jobs/Tasks/Permissions
QlikView Designers
Build QlikView application interfaces – no scripting
QlikView App Support
Support QlikView Apps and End Users
End Users
Users of QlikView applications

Strengths:

- all development costs born by departments
- flexible, fast development
- minimal central team training

Challenges:

- lack of consistency
- lack of re-use and standardization
- lack of governance
- lack of skill set sharing across departments
- higher TCO over time

Author: BPN, QlikTech NA