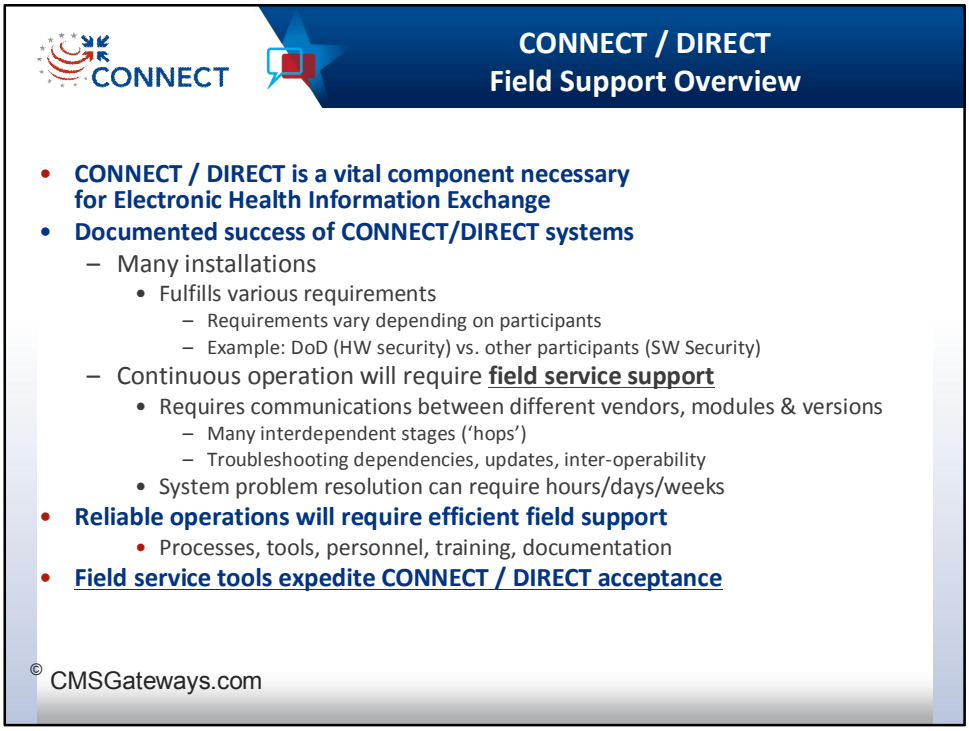

CONNECT
- Field Support Toolbox
- Debug procedures
Nick Booth
Technical Director
CMSGateways.com
(Content Management Systems)
FEDERAL HEALTH ARCHITECTURE
FHA


CONNECT
CONNECT / DIRECT
Field Support Overview
• CONNECT / DIRECT is a vital component necessary for Electronic Health Information Exchange
• Documented success of CONNECT/DIRECT systems
– Many installations
• Fulfills various requirements
– Requirements vary depending on participants
– Example: DoD (HW security) vs. other participants (SW Security)
– Continuous operation will require field service support
• Requires communications between different vendors, modules & versions
– Many interdependent stages ('hops')
– Troubleshooting dependencies, updates, inter-operability
• System problem resolution can require hours/days/weeks
• Reliable operations will require efficient field support
• Processes, tools, personnel, training, documentation
• Field service tools expedite CONNECT / DIRECT acceptance
© CMSGateways.com

## CONNECT/DIRECT Case study: CMS Electronic Report workflow

Health Care Provider

CMS

Quality Report PHI

Feedback



## CMS electronic report requirements

Health Care Provider

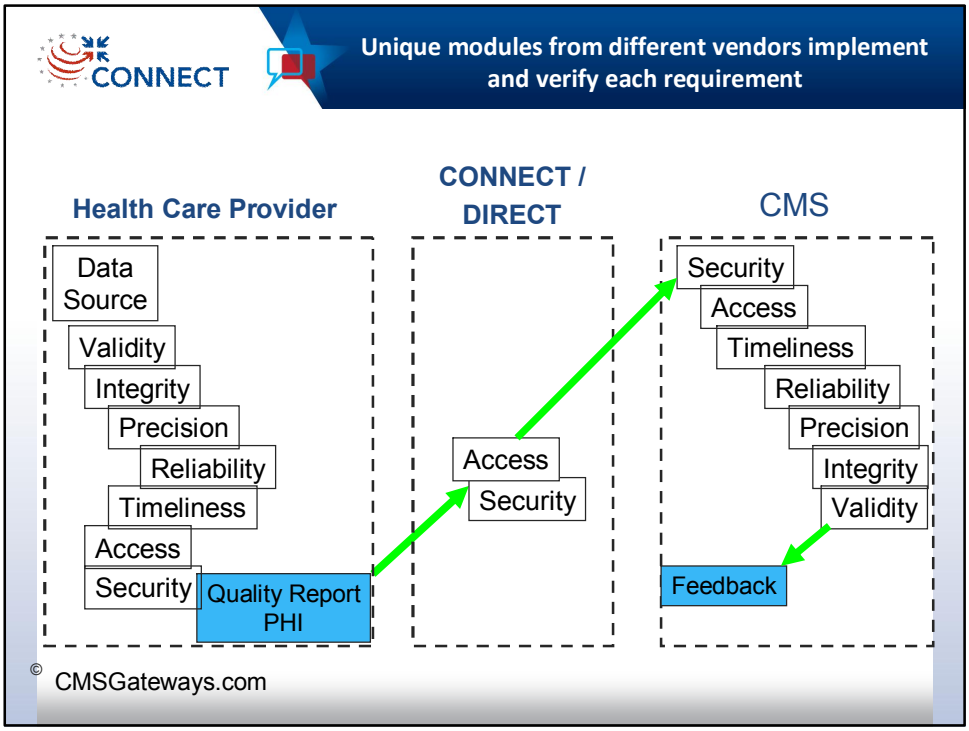
Quality Report PHI

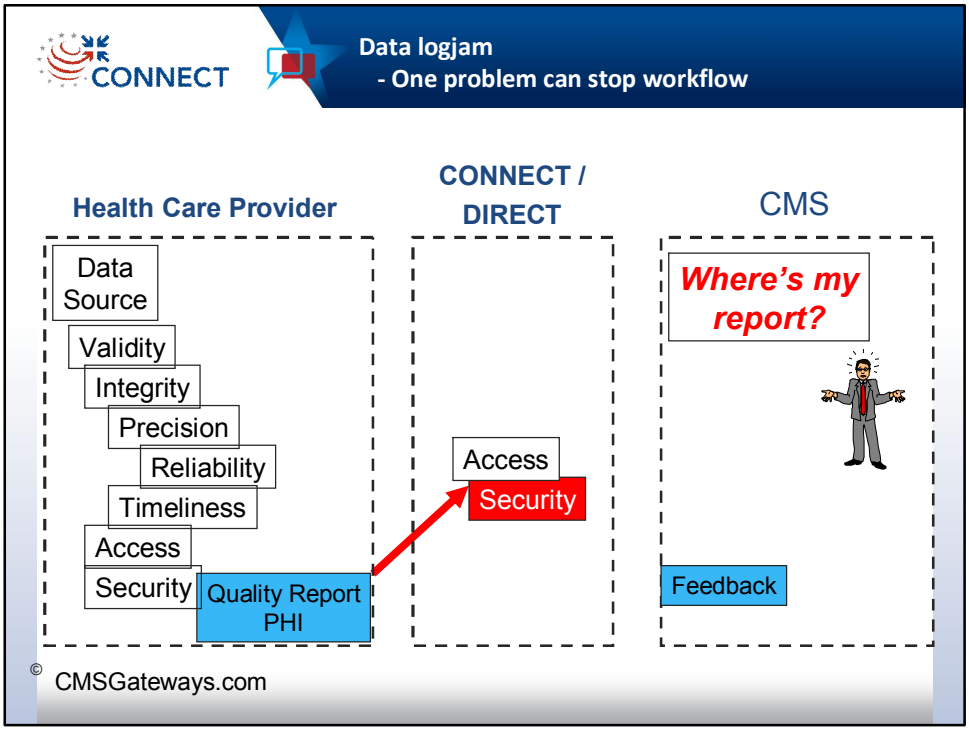
**CMS Reporting Requirements**

1. Validity
2. Integrity
3. Precision
4. Reliability
5. Timeliness
6. Access
7. Security

CMS

Feedback


CONNECT
Unique modules from different vendors implement and verify each requirement
Health Care Provider
CONNECT / DIRECT
CMS
Data Source
Validity
Integrity
Precision
Reliability
Timeliness
Access
Security
Quality Report PHI
Access
Security
Security
Access
Timeliness
Reliability
Precision
Integrity
Validity
Feedback
© CMSGateways.com


CONNECT
Data logjam
- One problem can stop workflow
Health Care Provider
CONNECT / DIRECT
CMS
Data Source
Validity
Integrity
Precision
Reliability
Timeliness
Access
Security
Quality Report PHI
Access
Security
Where's my report?
Feedback
© CMSGateways.com

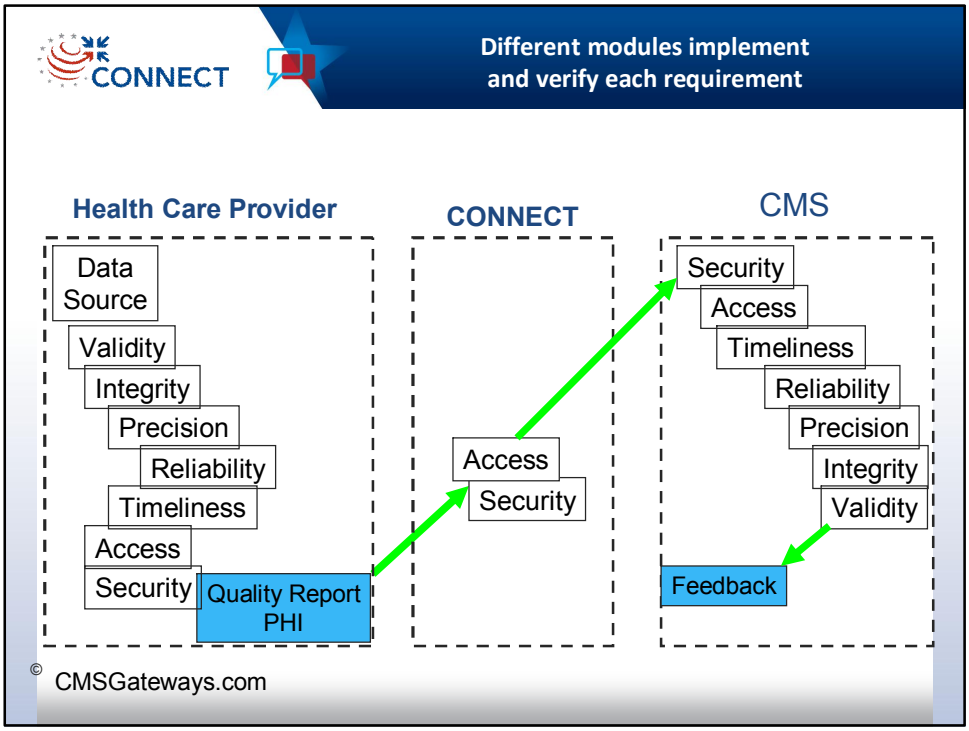
## CONNECT/DIRECT Field Support Overview

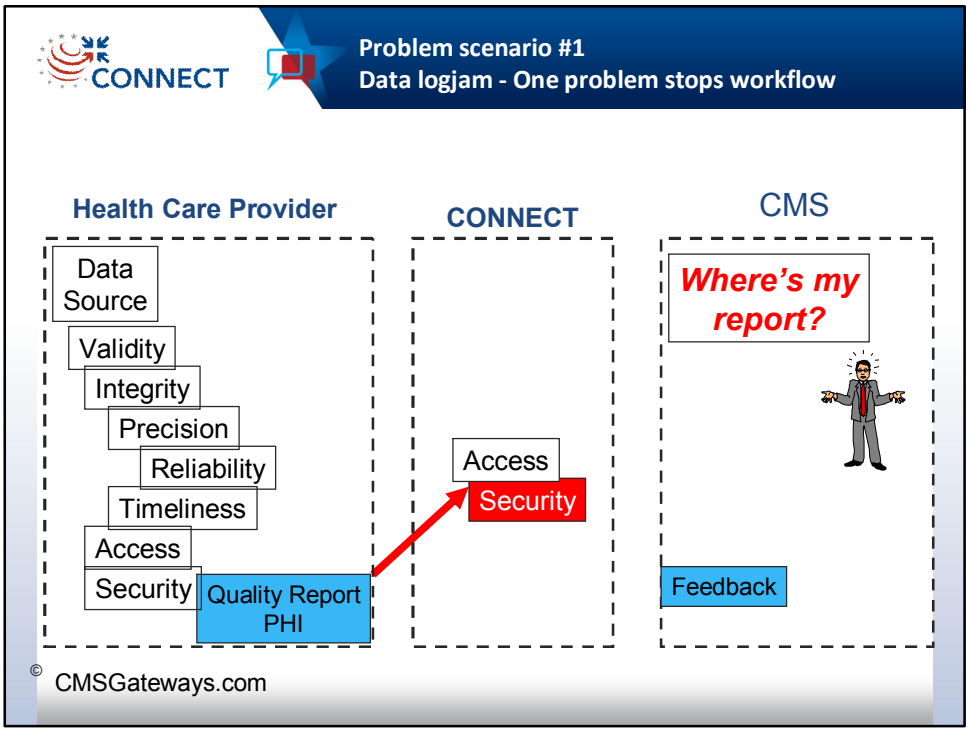
- **Current Problem Determination (PD) process characteristics**
  - Labor intensive diagnosis
    - Manually assemble, correlate, and interpret logs
    - Repetitive, time consuming problem resolution tasks
    - Advanced skills and extensive debug time (hours/days) required
- **System design has impact on PD**
  - Are PD diagnostics integrated into code paths?
  - **CONNECT 4.x has begun integration of PD logs & metrics!**
- **Poor problem determination processes & lack of PD tools lead to…**
  - **Increased cost of ownership**
  - **Decreased utilization**
  - **Decreased market share**
  - **Disconnected & mothballed technology**

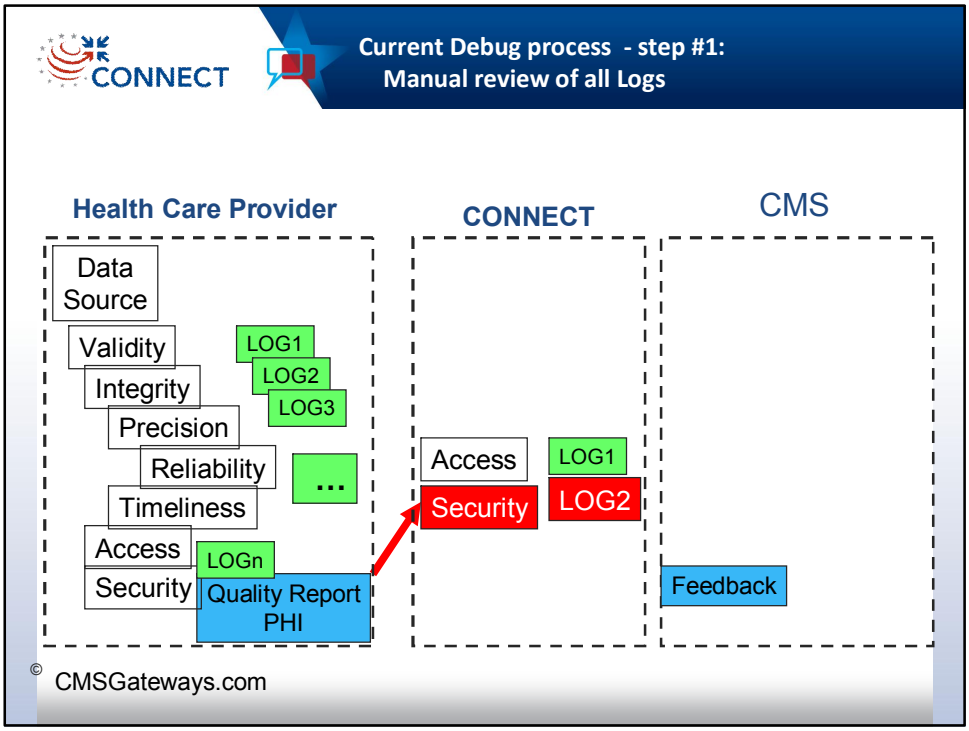


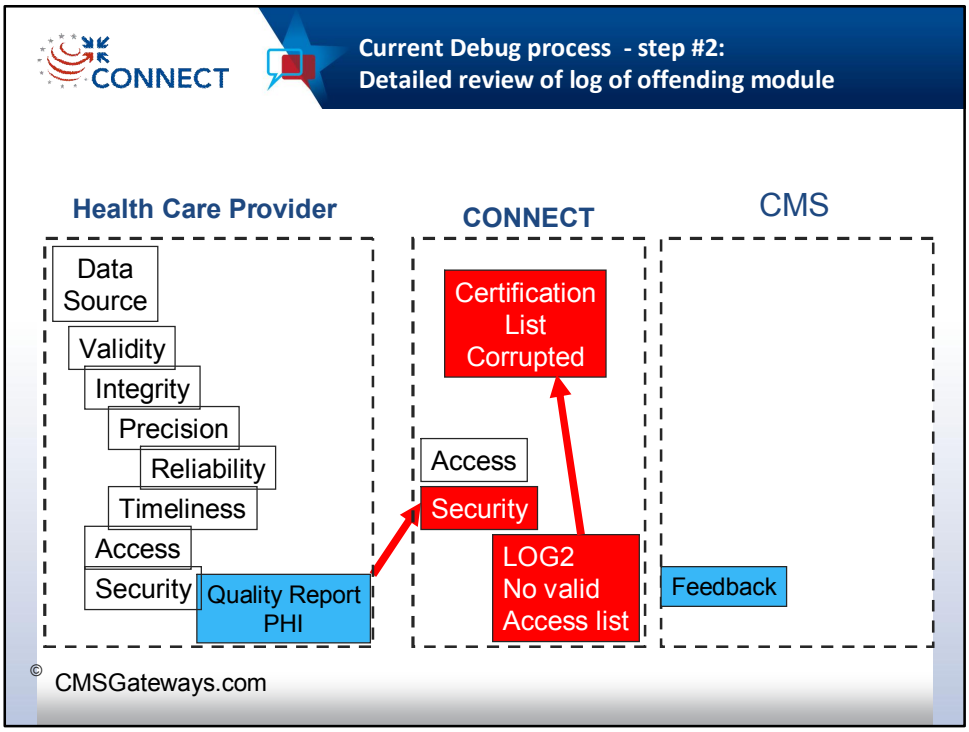
## CONNECT/DIRECT Field Support Overview

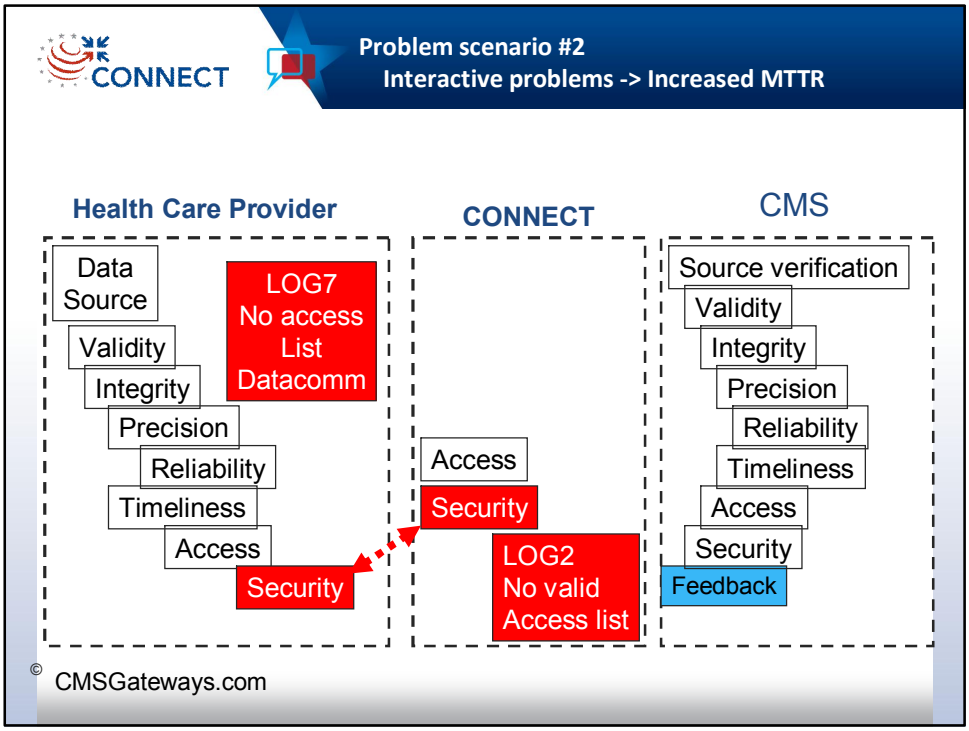
- **Field Support Goal: Improve maintainability**
  - Automated diagnostic tools – Reduced downtime
  - Streamlined diagnostic processes –Reduce cost of support
- **Components of maintenance:**
  - Reliability
    - Optimize MTBF (Mean Time Between Failure)
  - Availability
    - Total time a system is expected to function
    - Mean Time Before Repair (MTBR)
  - Serviceability
    - Ease of maintenance & repair
    - Minimize MTTR (Mean Time To Recovery/Repair)
  - **RAS** – Reliability, Availability, Serviceability

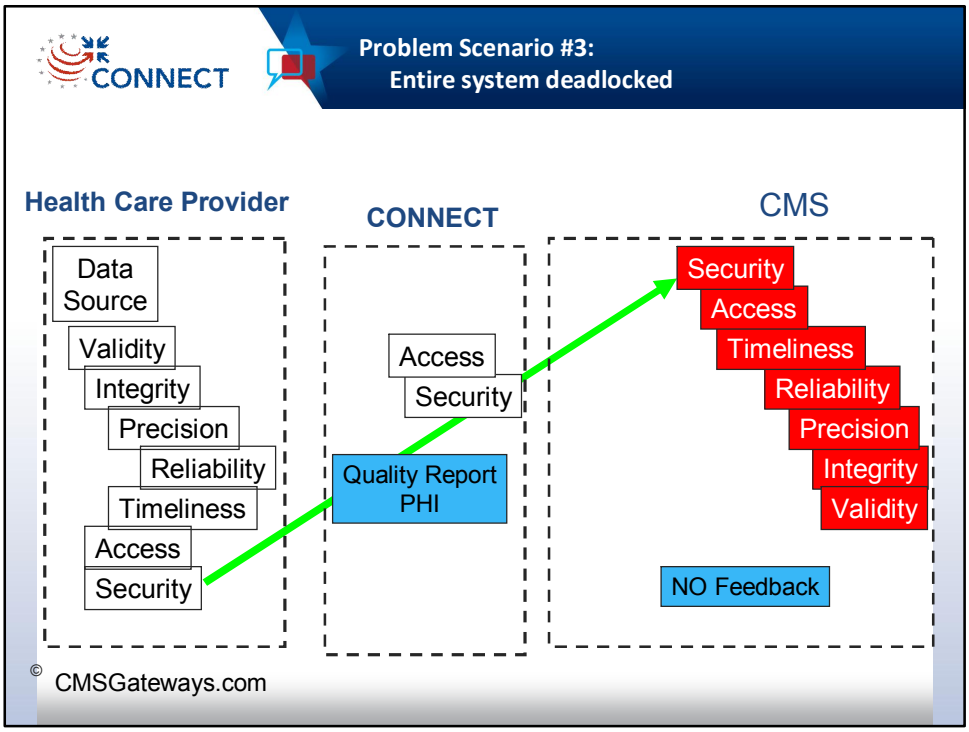

CONNECT
Different modules implement and verify each requirement
Health Care Provider
CONNECT
CMS
Data Source
Validity
Integrity
Precision
Reliability
Timeliness
Access
Security
Quality Report PHI
Access
Security
Security
Access
Timeliness
Reliability
Precision
Integrity
Validity
Feedback
© CMSGateways.com

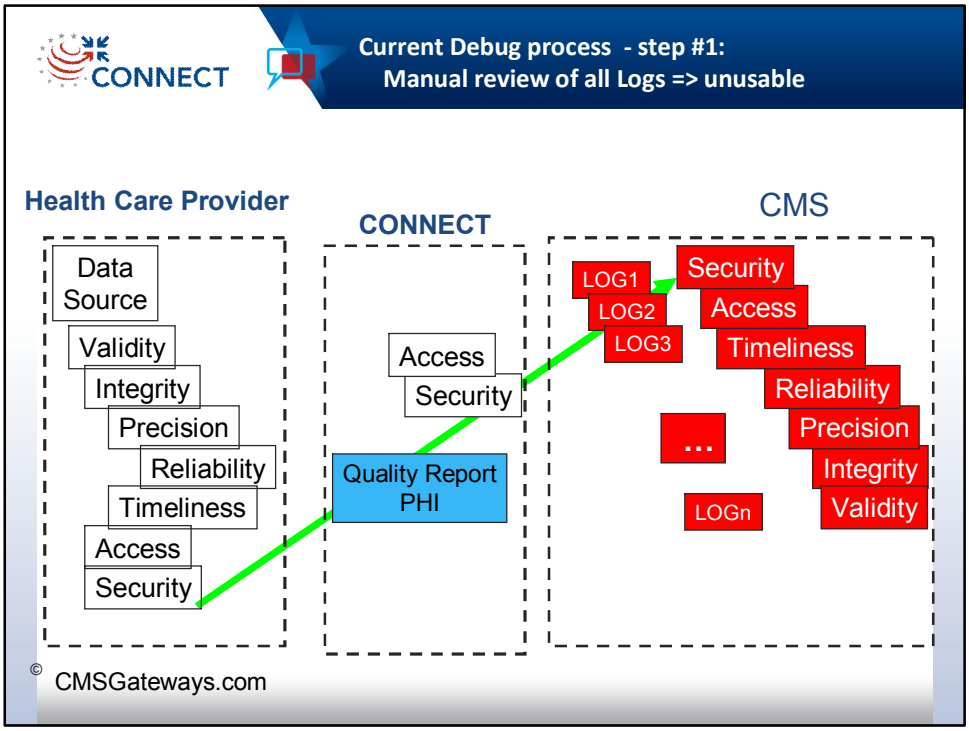

CONNECT
Problem scenario #1
Data logjam - One problem stops workflow
Health Care Provider
CONNECT
CMS
Data Source
Validity
Integrity
Precision
Reliability
Timeliness
Access
Security
Quality Report PHI
Access
Security
Where's my report?
Feedback
© CMSGateways.com

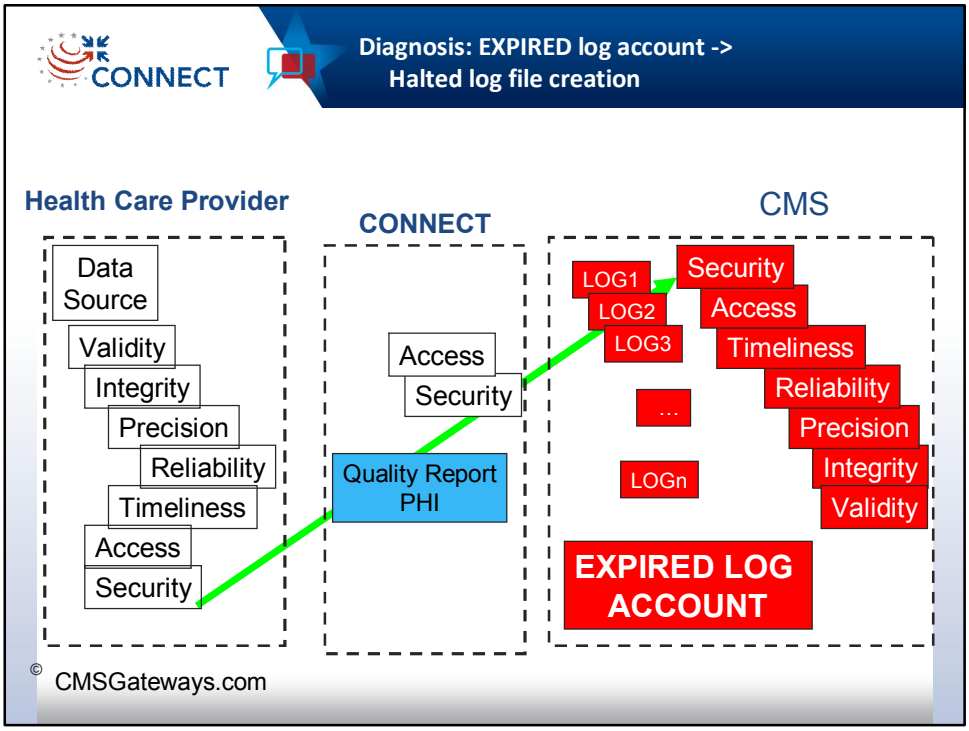

CONNECT
Current Debug process - step #1:
Manual review of all Logs
Health Care Provider
CONNECT
CMS
Data Source
Validity
Integrity
Precision
Reliability
Timeliness
Access
Security
LOG1
LOG2
LOG3
...
LOGn
Quality Report PHI
Access
Security
LOG1
LOG2
Feedback
© CMSGateways.com


CONNECT
Current Debug process - step #2:
Detailed review of log of offending module
Health Care Provider
CONNECT
CMS
Data Source
Validity
Integrity
Precision
Reliability
Timeliness
Access
Security
Quality Report PHI
Certification List Corrupted
Access
Security
LOG2
No valid Access list
Feedback
© CMSGateways.com


CONNECT
Problem scenario #2
Interactive problems -> Increased MTTR
Health Care Provider
CONNECT
CMS
Data Source
Validity
Integrity
Precision
Reliability
Timeliness
Access
Security
LOG7
No access List
Datacomm
Access
Security
LOG2
No valid Access list
Source verification
Validity
Integrity
Precision
Reliability
Timeliness
Access
Security
Feedback
© CMSGateways.com


CONNECT
Problem Scenario #3:
Entire system deadlocked
Health Care Provider
CONNECT
CMS
Data Source
Validity
Integrity
Precision
Reliability
Timeliness
Access
Security
Access
Security
Quality Report PHI
Security
Access
Timeliness
Reliability
Precision
Integrity
Validity
NO Feedback
© CMSGateways.com


CONNECT
Current Debug process - step #1:
Manual review of all Logs => unusable
Health Care Provider
CONNECT
CMS
Data Source
Validity
Integrity
Precision
Reliability
Timeliness
Access
Security
Access
Security
Quality Report PHI
LOG1
LOG2
LOG3
...
LOGn
Security
Access
Timeliness
Reliability
Precision
Integrity
Validity
© CMSGateways.com


CONNECT
Diagnosis: EXPIRED log account ->
Halted log file creation
Health Care Provider
CONNECT
CMS
Data Source
Validity
Integrity
Precision
Reliability
Timeliness
Access
Security
Access
Security
Quality Report PHI
LOG1
LOG2
LOG3
...
LOGn
Security
Access
Timeliness
Reliability
Precision
Integrity
Validity
EXPIRED LOG ACCOUNT
© CMSGateways.com

## CONNECT/DIRECT Field Support Overview

- **Problem Determination (PD) components**
  - Problem management discipline
    - Automate Maintenance functions
      - Identify RAS tools requirements (Reliability, Availability, Serviceability)
  - PD workflow procedures
    - PD query process
    - PD environments
  - RAS tool solutions
    - Open source vs. proprietary
    - Diagnostic information from variety of sources



## CONNECT/DIRECT Field Support Objective

- **Problem Management Discipline**
  - Problem Documentation:
    - Confirm, categorize, prioritize & publish
  - Acquire relevant Problem Determination (PD) data
    - Automate common PD support tasks
    - Involve all participants: Users, field support staff, 3rd parties
      - Example: Xref problems lists from other bugs & third party modules
    - Apply tools => observe & control system
  - Expedite the identification of fault source(s)
    - PD data analysis (Dev team, test team or Field support)
      - Transform intermittent bug => regular bug
        - » Resolve the mystery cause(s)
  - Implement Bug fix (w/ no side effects)

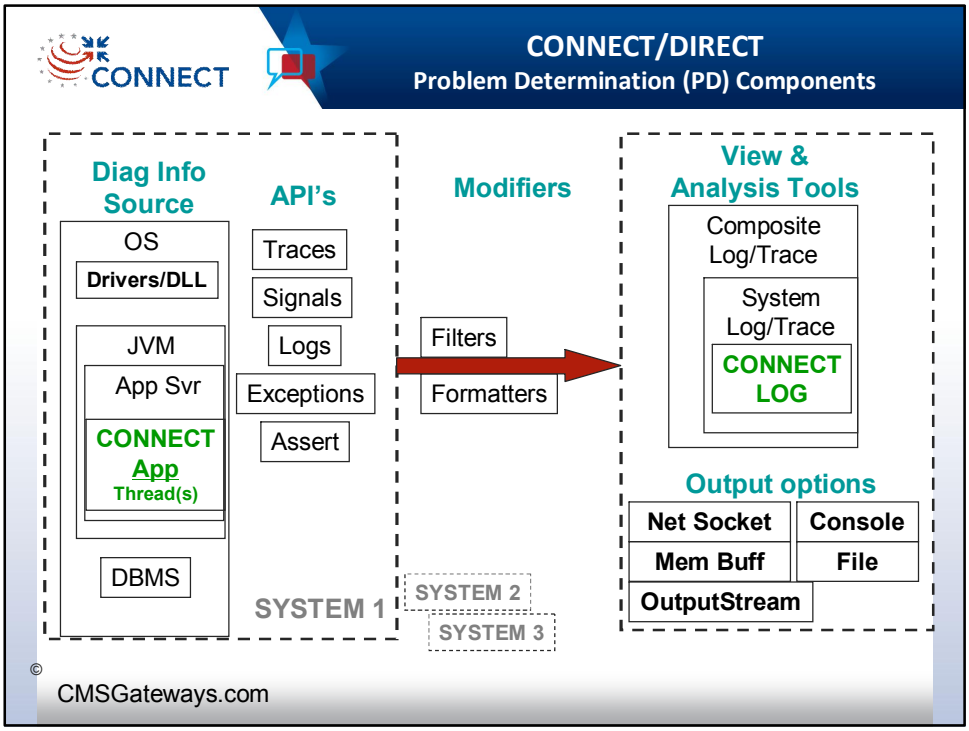
## CONNECT/DIRECT Field Support Workflow

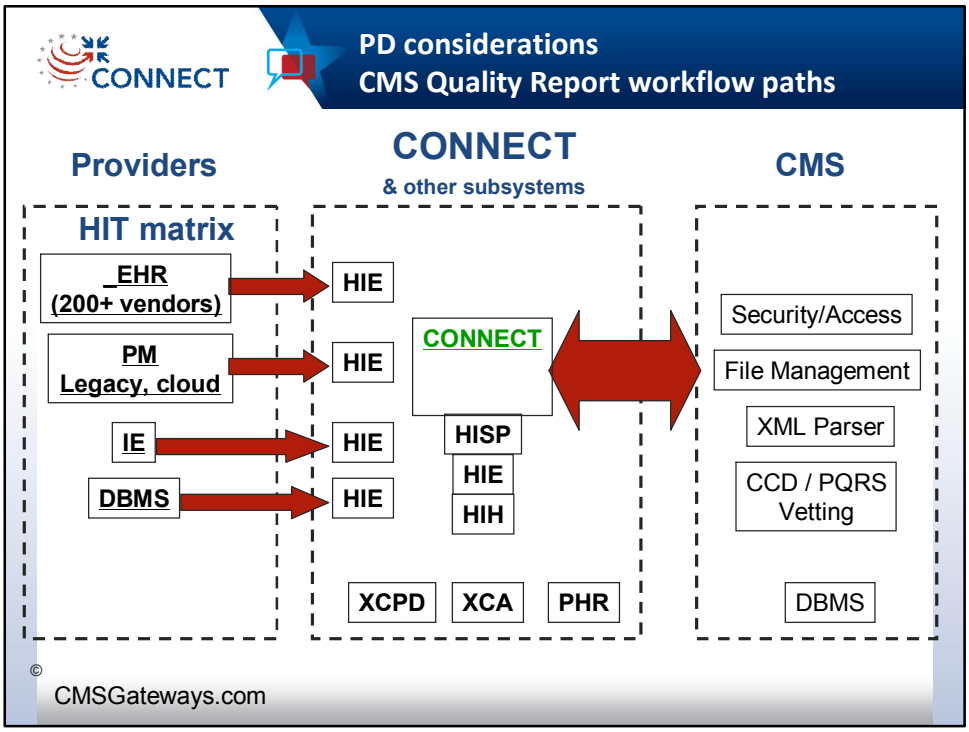
- **Diagnostic workflow procedures**
  - Goal: Acquire relevant diagnostic data
  - Understand operations
    - Cartography - Functional map of complete system
    - Internals: Modules & data flow
    - Externals: Protocols & states of transaction
  - Configuration, version control
    - Standardized update procedures
    - Module interdependencies
  - Tools and Diagnostic data acquisition processes
  - Extend development & test bench into field
    - Enable Users & Field personnel to collect USEFUL diagnostics



## CONNECT/DIRECT Field Support Tools

- **Problem Determination (PD) automation tools**
- **Automated data collection**
  - Configuration, Input/output, status, version
    - Heterogeneous environment – modules & subsystems
  - Diagnostic API's: Logs, traces, events, signals, exceptions
- **Forensic data mining**
  - Log merge, parsing, sorting & analysis
    - Identify events leading up to problem
    - Isolate source(s) of problems


CONNECT
CONNECT/DIRECT
Problem Determination (PD) Components
Diag Info Source
API's
Modifiers
View & Analysis Tools
OS
Drivers/DLL
JVM
App Svr
CONNECT App
Thread(s)
DBMS
Traces
Signals
Logs
Exceptions
Assert
Filters
Formatters
Composite Log/Trace
System Log/Trace
CONNECT LOG
Output options
Net Socket
Console
Mem Buff
File
OutputStream
SYSTEM 1
SYSTEM 2
SYSTEM 3
©
CMSGateways.com


CONNECT
PD considerations
CMS Quality Report workflow paths
Providers
CONNECT
& other subsystems
CMS
HIT matrix
_EHR (200+ vendors)
PM Legacy, cloud
IE
DBMS
HIE
HIE
HIE
HIE
CONNECT
HISP
HIE
HIH
XCPD
XCA
PHR
Security/Access
File Management
XML Parser
CCD / PQRS Vetting
DBMS
©
CMSGateways.com

## Problem Determination (PD) Queries

- **Problem Determination Workflow procedures**
- **PD queries**
  - Accurate problem report?
  - Different system?
  - Different state?
  - Different data?
- **Complete problem report via PD queries**
  - User interview
  - Diagnostic data acquisition PD procedures



## Problem Determination (PD) Query #1

- **Is this problem report / observation accurate?**
  - Corrupted problem record
    - Incomplete, unreliable communications
  - Misattribution / false correlation
    - Intermittent problem misconstrued => non-intermittent problem w/complex and unlikely set of causes (MSWord=>Win crash)
  - Misrepresentation
    - Incomplete assessment (PS3 malfunction, hidden connector was unplugged)
  - Different operators have different problem tolerances and sensitivities
    - Sensitivity and vary with time of day
  - Irrelevant problem (i.e. Observation is too accurate )

## Problem Determination (PD) Query #1

- **PD information categories - problem reports**
  - Timestamp, PD environment, priority, classification, scope of problem
  - Log augmentation: Track multiple entries by multiple authors



## Problem Determination (PD) Query #2

- **Is it a different system?**
  - Automatic or IT updates
  - Trespassing system - foreign intrusions
  - Configuration changes
    - Third party add-ons affect code paths
    - Drivers, driver stacks, DLL's, apps, monitors
  - Documentation & processes in place
    - Automated version comparison / control programs
    - Rollbacks & version control co-ordination
      - Third parties
      - Documented version inter-dependencies

## Problem Determination (PD) Query #3

- **Is system in a different state?**
  - System in different mode?
    - User or protocol may have set different mode
  - Improper init
    - Changes in config, registries, resources & routing tables
  - Resource denial
    - File, stream, or other resource
    - Corrupted, does not exist, locked by another process/thread
  - Occasional functions
    - Auto-save, periodic maintenance, internal garbagecollect
  - Progressive data corruption (timing loops, rounding)
  - Progressive destabilization
    - Destabilizing event – create wild pointer
    - Initiating event – Use wild pointer



## Problem Determination (PD) Query #4

- **Did system receive different data?**
  - Secret / different boundaries and conditions
    - Software may act differently in different parts of input space
    - Different logic invoked by chosen option(s)
  - Input corruption
    - Inputted corrupted or intercepted
    - Deus ex machina - Third party influence
      - Fellow developer/tester, other user, hacker
    - Accidental or Ghost input –
    - Signals from different peripherals, network
      - » sun => Optical mouse
      - » RTF from MS Word & MS Wordpad are not the same
  - Consider time & loading as an input

## Problem Determination (PD) Processes

- **PD Environments**
  - Development, System Test, Multi-System Test, Field Install
- **PD Tools**
  - Scope of diagnostic data
    - Systemwide, Server, Application, Module
    - Component interactions
  - Tool providers: Open Source & Proprietary
- **Setup communications between all of the above!**



## Problem Determination (PD) environment #1 Software Development

1. **Software Development environment**
   - Interactive Debugging - IDE / Eclipse (or ?)
     - Call stack, variables values, Breakpoints
     - Printf debugging / TRON
     - ASSERT
     - Post-Mortem Debug – crash analysis
     - Semantic errors - Static code analysis tools

## Problem Determination (PD) environment #2
## System test suite

### 2. System test suite environment

- Purpose: Decrease costs of functional defects
  - Each Development stage has associated defect resolution costs
    - Requirements, Arch, Construction, System test, Post release
  - Defect costs more if caught at later stage
    - Field Support => multiple updates => configuration changes
    - Cloud/Continuous deployment reduce costs of later stages
- Test Input combinations and preconditions
  - Automated finite combinational tests
  - Get greater test coverage with fewer tests
  - Compromise test speed vs. test depth
- Need coverage of non-functional attributes
- Usability, scalability, performance, compatibility (version), reliability



## PD environment #3
## Inter-system bench test

### 3. Inter-system bench test

- Controlled environment
  - Version, loading, data mix
- Multi-vendor, multi-module
  - Multiple overlapping errors increase PD complexity
- Controlled debugging
  - Dedicated 'offline' systems => remote test bed
- Problem determination
  - Balance performance with Serviceability (RAS)
  - Automated data collection
  - Test offline analysis procedures - automated & manual

## PD environment #4 Field Install

### 4. Customer Install - Field Service

- Uncontrolled environment
  - Version, loading, data mix
- Multi-vendor, multi-module
  - Multiple overlapping errors increase PD complexity
- Online, live debugging
  - #1 Goal of Field Support – Keep system online!
  - Can dedicate extra system as remote test bed
- Problem determination
  - Balance performance with Serviceability (RAS)
  - Automated data collection
  - Offline analysis - automated & manual



## PD debug mode #1 => Source debug

### 1. Logic debug of an app module

- Hard faults - "ASSERT" –
  - Usually removed from production code
- Intermittent problems
  - Stress system to recreate problem
  - If race condition exists, usually affected by debug process
    - Threading ,memory management issues
    - Debugger affects timing, can exaggerate or solve problem.
  - Fuzz tests w/random input => irrational border cases

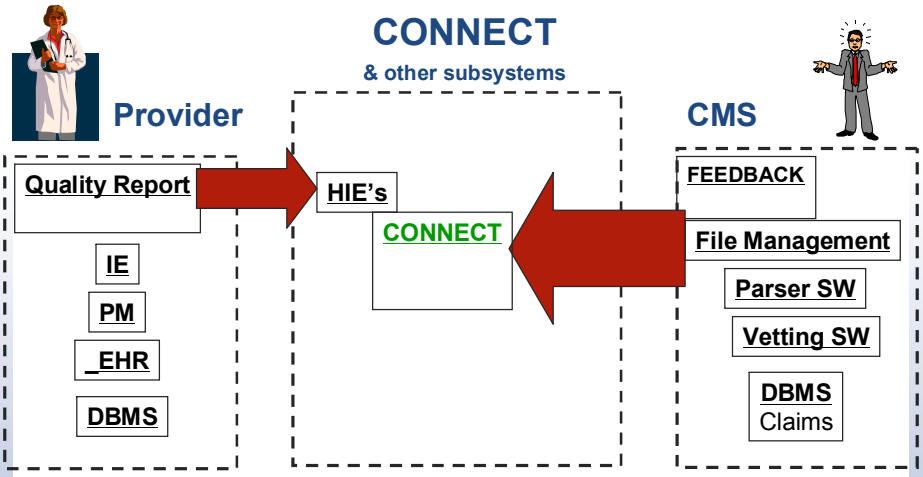
## PD debug mode #2 – API debug

**2. Problems between system components**

- Heterogeneous environment
  - Must track version history of (related) subsystems
    - Inter-Dependencies
  - Scripted automated compare – look for version delta
- Automated test scripts
  - Version dependencies – Example: NwHIN protocols
  - Options
    - Race conditions – Test configurations => vary timing
    - System loading – Test configurations => vary sources, sinks & data loads



## Inter-System Datacomm PD



## PD debug mode #3 - Communications

### 3. Communication protocols between systems

- PD Transaction Analysis
  - Between CONNECT and trading partners such as….
  - NIST: Conformance testing against a reference
  - Other vendors:Interoperability (@ IHE connectathon)
- CONNECT V4.0 incorporates PD Metric & Error Logs
  - Performance
  - Transaction Type, Payload
  - Error Messages log
- XDS.b Transaction/datacomm tools & reference materials
  - ihe-xds-implementors@googlegroups.com
  - NIST Test Tools -> http://hit-testing.nist.gov:12080/xdstools2
  - Connectathon: http://www.ihe.net/connectathon/



## PD debug mode #4 – Security

### 4. Security Management problems

- **CERT management a time consuming debug issue!**
  - Default certificate configuration
  - Obtaining signer certificate from a remote port
  - Remote signer certificate retrieval
  - Validating a remotely-retrieved signer certificate
  - Replacing certificates and signers
  - Certificate expiration monitor and dynamic run time updates
  - Advanced certificate and key management issues
  - CERT management tools
    - Websphere GUI admin console
    - Windows command line => certmg.exe

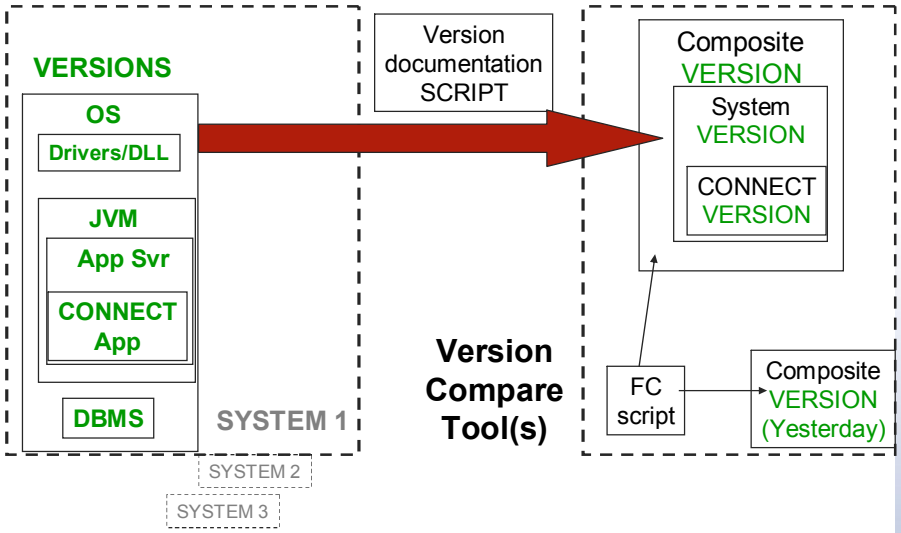
## PD debug mode #5 – Intermittent bug

**5. Field Multi-System Intermittent problems**

- Field Support procedures & tools requirements
- Support Multi-vendor environments
  - Version dependencies of multiple modules
  - Disparate data sources
- Automated data collection
  - Minimize expertise required for data acquisition
- Automate module / code path analysis
  - Offline analysis merges diag data from different sources
- Minimize and localize Performance tradeoffs
  - Serviceability (RAS) AND
  - System loading, throughput, stability



## PD Doc #1
## Automated Version Documentation!



## PD Doc #1 - System config docs

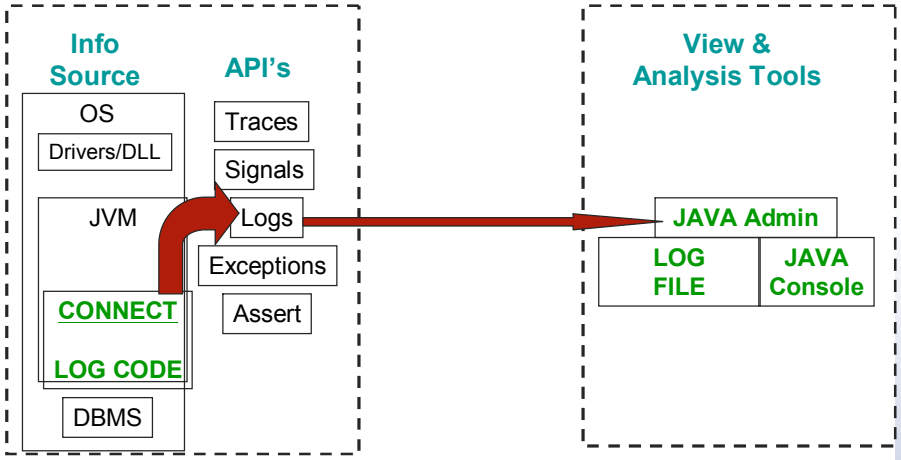
- **System DOCUMENTATION**
  - Timely automated gathering of CONFIG
    - Modules / subsystems / OS
      - ALL VENDORS!
    - Date, time, checksums
  - Automated, scripted comparison
    - Establish Version / Change history
    - Immediately spot any delta's
    - Helps to map out updates, rollbacks, hotfixes, etc.
    - Some people rely on dump/trace/log for same info
      - Delta's are not easy to extract and compare



## PD Doc #2 - Application Logs

- **Instrument your code!**
  - Log statements
- **Log data categories**
  - Performance counters ( system loading )
  - Stack traces
  - Race conditions ( timeout counters )

## PD Doc #2A – App Log via JVM






## Java JVM Log

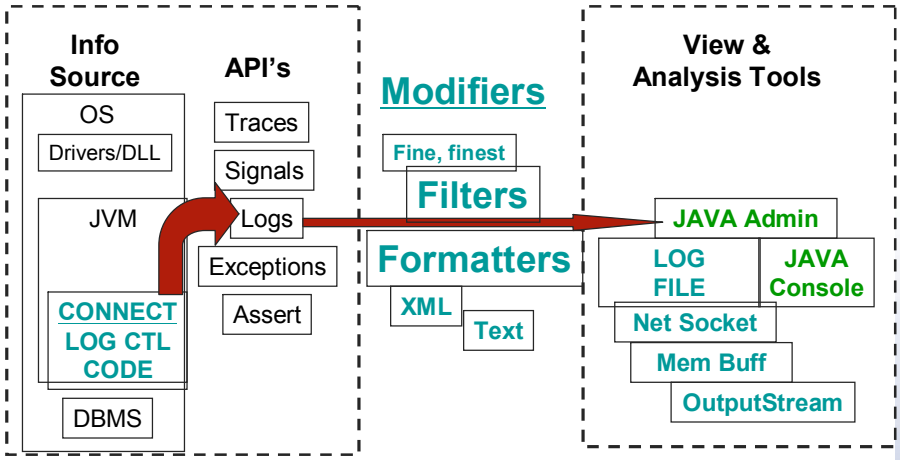
- **Logging**
  - redirect Java Console output to log file via Java Logging API.
- **To enable logging perform the following actions:**
  - Open Java Control Panel / Admin panel
  - Click Advanced tab.
  - Select Enable Logging under the Debugging option

## Java Log options

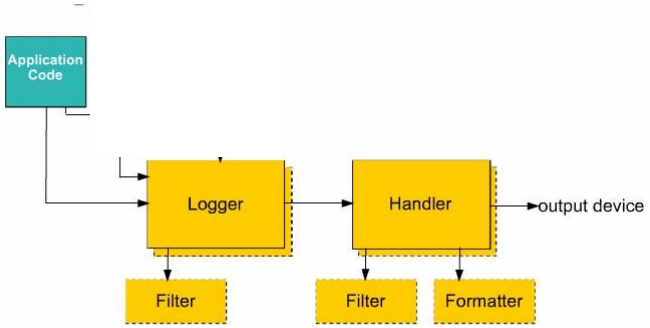
- **Options:**
  - Redirect system.out & system.err
    - To log file
    - To network socket
    - To Outputstream
    - To mem buffer
  - Rotating Log files
- **Formatters**
  - XML or Text
- **Levels:**
  - Severe, warning, info, config, fine, finer, finest



## App Log control (>JDK 1.4)

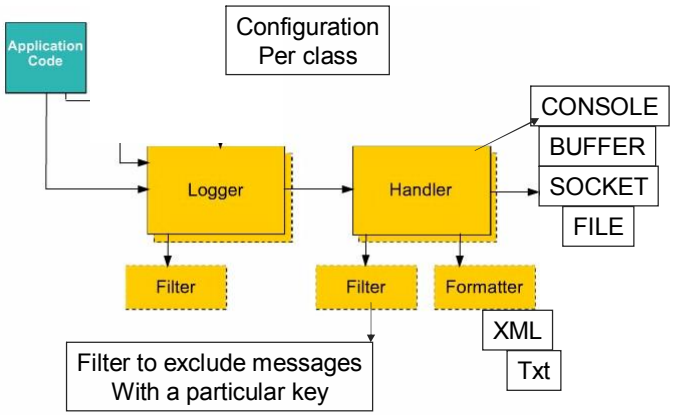


## JAVA Logging Framework






## Native JVM log components - functions

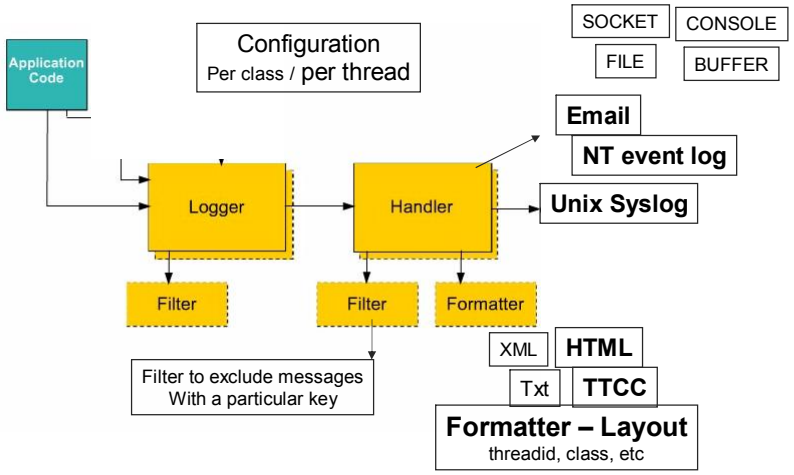


## More options – Open Source log4J

- **Sun Java Log API**
  - Universal
    - No external dependencies
    - Generally included in proprietary
- **log4J – Log API**
  - IBM ported RAS code => Java => Open Source
  - More output options
  - Flexible config
  - Longer history, smaller footprint, faster, thread safe



## log4J – More output options

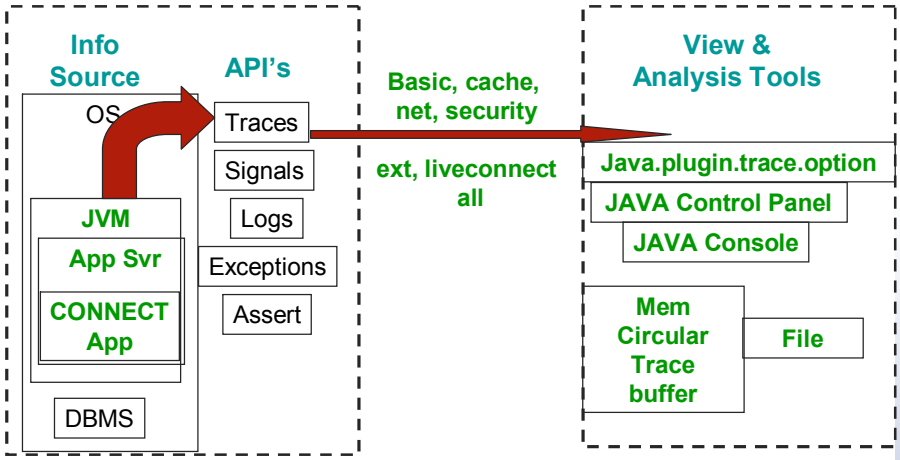


## Other log4J Log improvements

- **Improved Performance**
  - Asynchronous loggers
    - 10x throughput and orders of magnitude lower latency
- **Support for multiple APIs**
  - SLF4J – Simple logging façade
    - USER plugs in log framework at deployment time
  - Commons Logging
    - Change logging implementation without recompilation
- **Automatic Reloading of Configurations**
    - Without losing log events while reconfiguration is taking place.



## (PD) Mechanisms – JVM Trace



## Java Trace

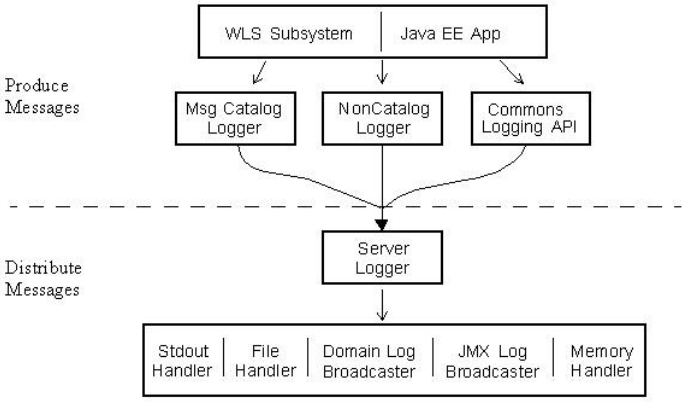
- **Set initial trace level for Java Web Start application**
  - Change trace level with API, trigger events
    - JVMRI (IBM - RAS Interface, deprecated)
    - JVMPI (Sun – Profiling interface, deprecated)
    - JVMTI (JVM / Oracle / IBM – Tools interface, current)
- **Set the deployment property deployment.trace.level.**
  - Basic, cache, net, security, ext, liveconnect, all



## Problem Determination Solutions

- **Open source PD**
  - Example: log4J
  - Advantages:
    - Source available for debugging/extensions
    - Small scale projects
    - Can be customized to emulate proprietary functionality
- **Proprietary PD**
  - System examples: Websphere, WebLogic
  - Advantages
    - Subsystem integration & testing – version control
    - PD tools => problem determinations cover more system components

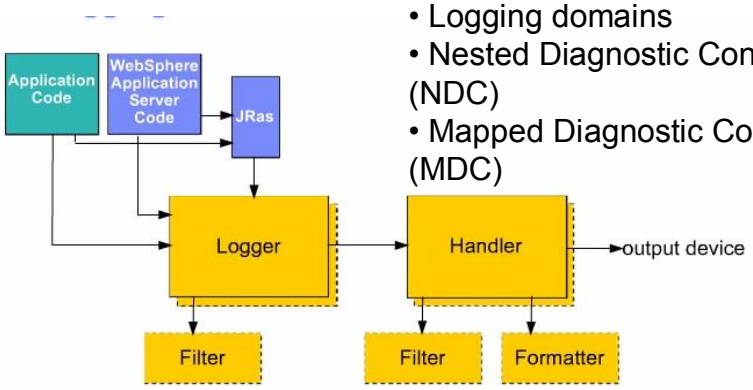
## WebLogic Log Diagram






## IBM Websphere LOG extensions

IBM extensions of log4J
- Logging domains
- Nested Diagnostic Contexts (NDC)
- Mapped Diagnostic Contexts (MDC)



## Advantages - Proprietary Solutions

- **IBM Websphere**
  - JVM log + log4J + proprietary extensions
- **Integrate Mainframe experience**
  - Streamlined binary log/trace 3x faster
  - Multi-Server Log merge
  - Advanced Filtering and Admin consoles
  - Merged Open source with proprietary extensions



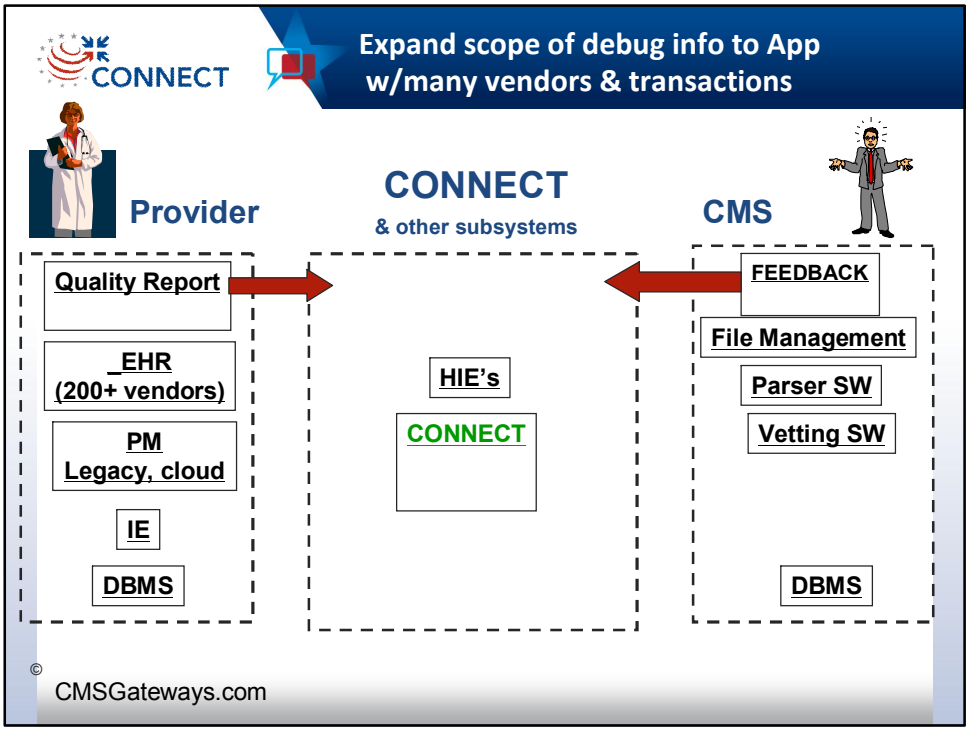
## Expand scope of debug info to App

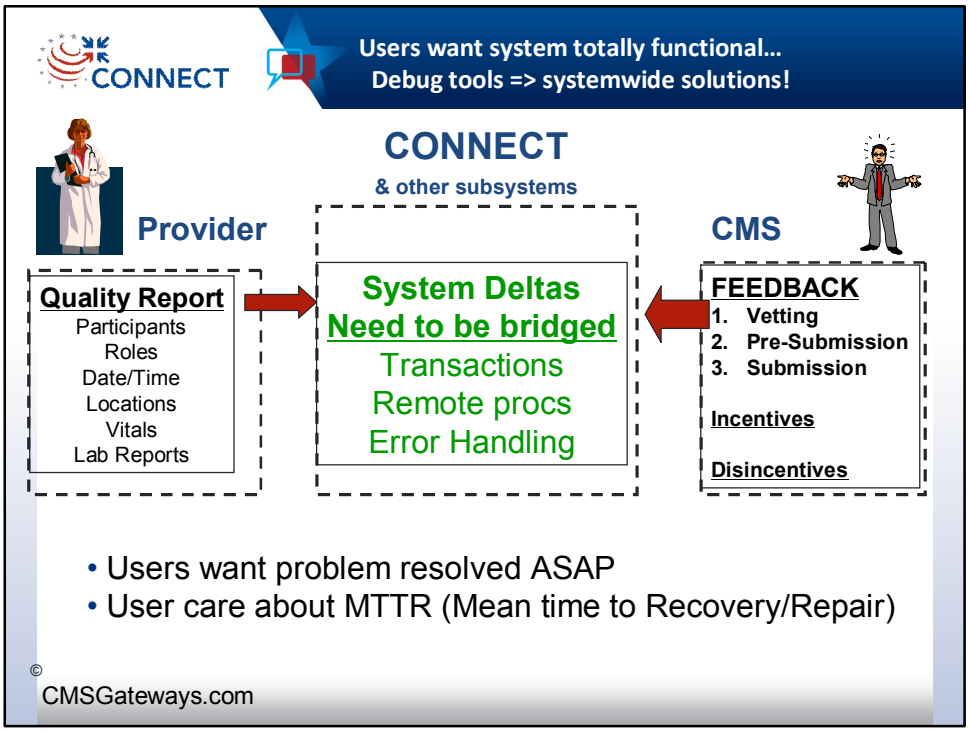
Health Care Provider

CMS

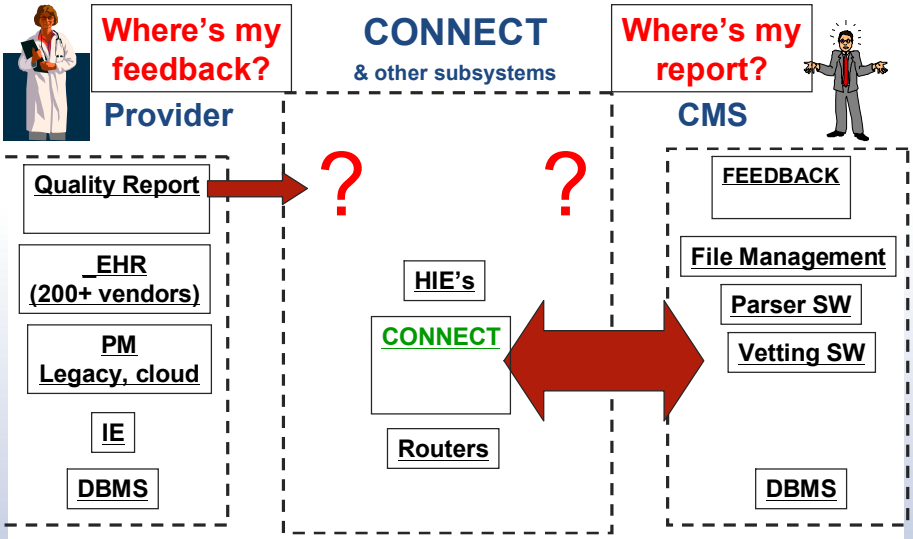
Quality Report (PQRS) PHI - XML

Feedback


CONNECT
Expand scope of debug info to App w/many vendors & transactions
Provider
CONNECT
& other subsystems
CMS
Quality Report
_EHR (200+ vendors)
PM Legacy, cloud
IE
DBMS
HIE's
CONNECT
FEEDBACK
File Management
Parser SW
Vetting SW
DBMS
© CMSGateways.com


CONNECT
Users want system totally functional…
Debug tools => systemwide solutions!
Provider
CONNECT
& other subsystems
CMS
Quality Report
Participants
Roles
Date/Time
Locations
Vitals
Lab Reports
System Deltas
Need to be bridged
Transactions
Remote procs
Error Handling
FEEDBACK
1. Vetting
2. Pre-Submission
3. Submission
Incentives
Disincentives
• Users want problem resolved ASAP
• User care about MTTR (Mean time to Recovery/Repair)
© CMSGateways.com

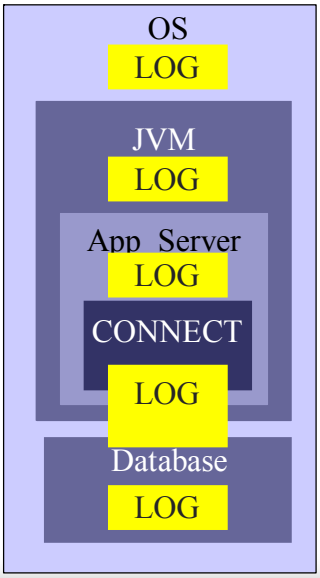
## Tools must be able to identify the many sources of system fault(s)





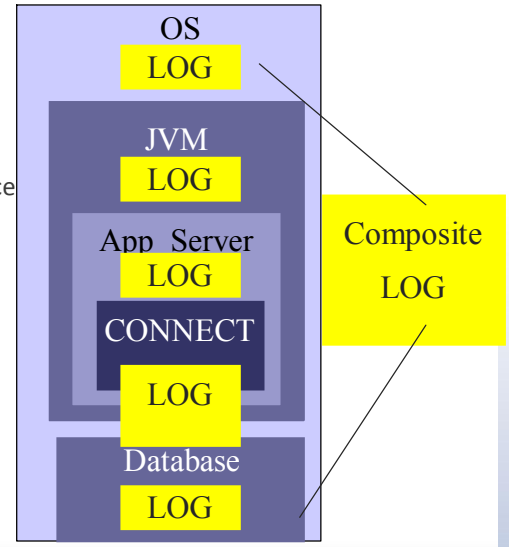

## Each subsystem has diagnostic logs

- **Multiple logs**
  - System wide vs. app specific
  - Defined interfaces
  - Improve code maintenance
- **Scope of diagnostics**
  - System wide vs. app specific
  - Defined vs. custom interfaces
- **Tradeoffs**
  - Interaction
    - Other system components
    - Other Apps
    - External systems
  - Impact on system performance
  - Communications ability



## Need for composite logs

- **Multiple log functions**
  - Sync and parse
  - System wide &. app specific
  - Defined interfaces
  - Improve SYSTEM maintenance
- **Scope of diagnostics**
  - System wide
  - All interfaces

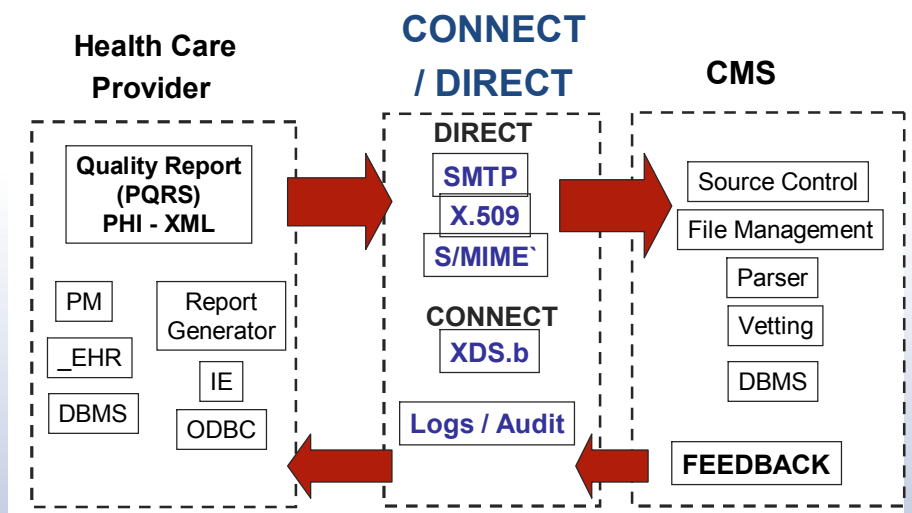





## System Support - delegation

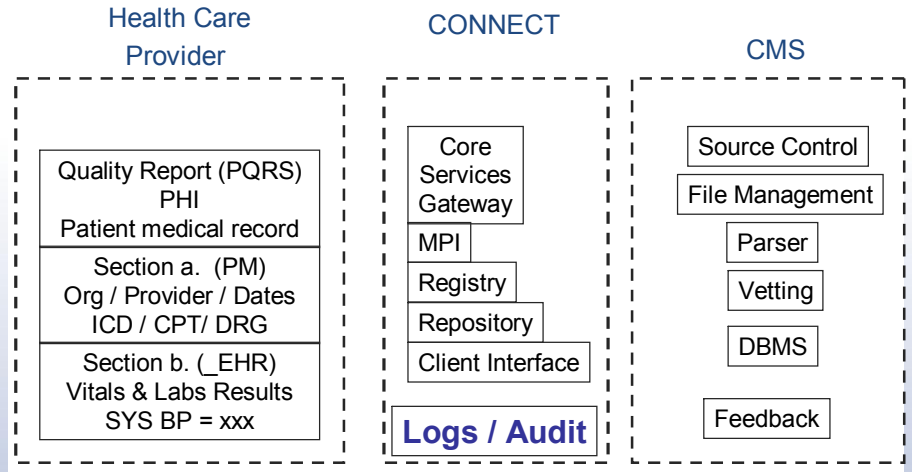
- **Handoff of system support**
  - From Programmers to Field support
  - Planned transition
  - Enable programmers to be more efficient
- **CONNECT Improvement**
  - RAS – Reliability, Availability, Serviceability
  - (Semi) automatic problem resolution
  - System Modularity

## CMS report pathways (2014/2015)






## CMS report components



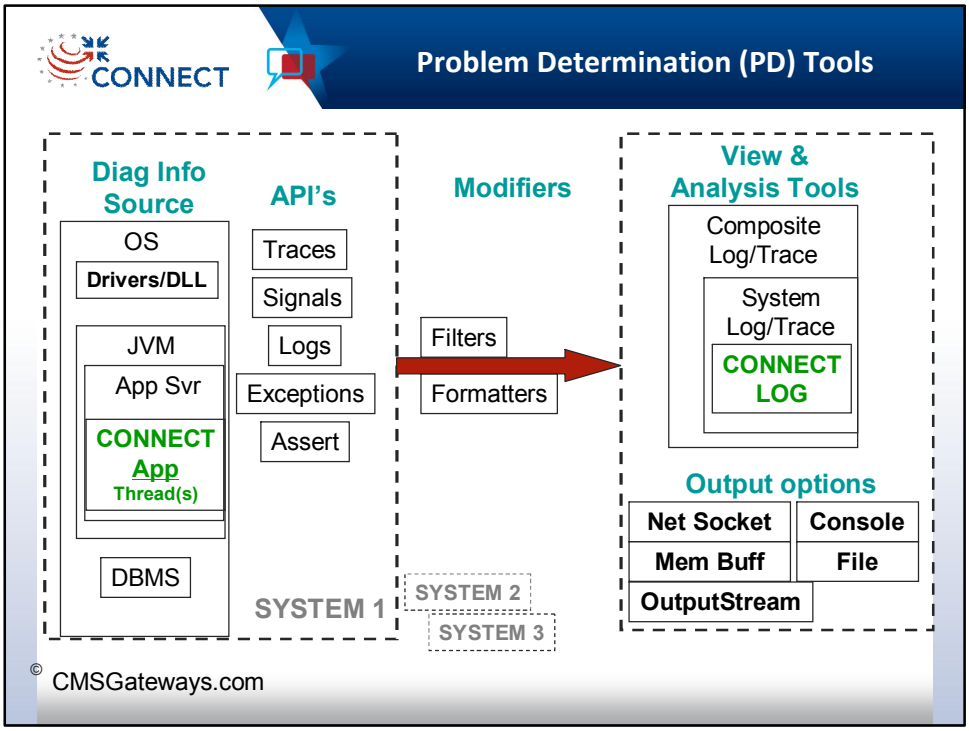
## Review CONNECT Field Support

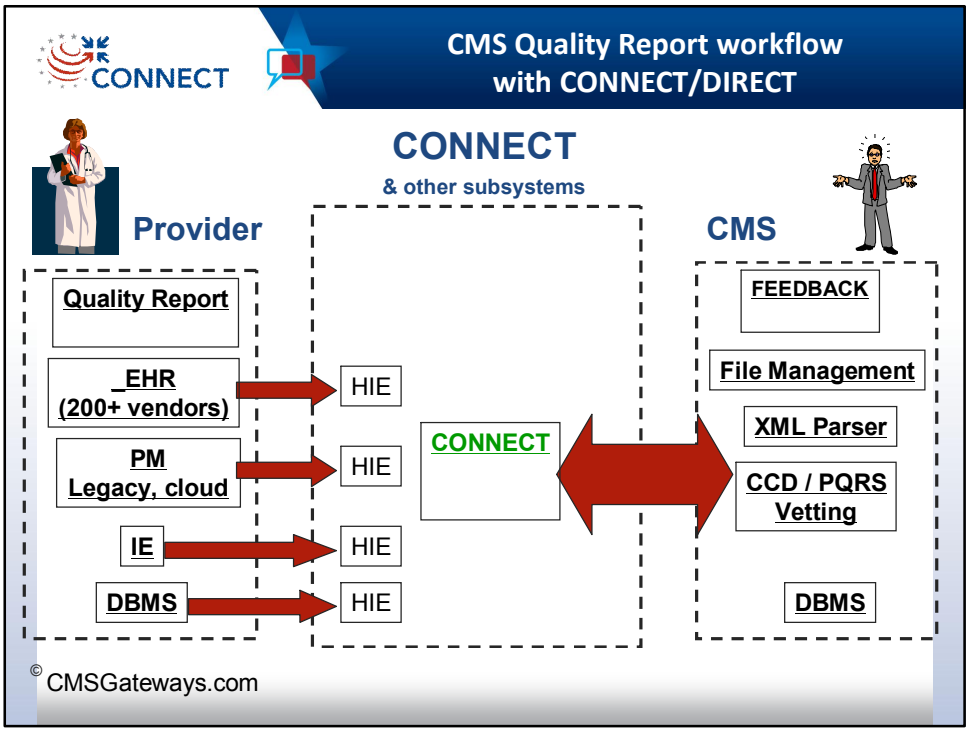
- **Coordinated Problem Determination (PD)**
- **Goal: Improve RAS**
  - Increase Reliability, Availability, Serviceability
- **Milestones to goal**
  - Problem management discipline
  - Problem determination workflow procedures
  - RAS tool solutions
    - Open source & Proprietary
    - Vendor choice(s) affects procedures, staffing & MTTR
      - MTTR (Mean Time To Recovery/Repair)



## Review CONNECT/DIRECT PD processes

- **Standardized Field Support RAS procedures**
  - Enable field support and non-programmers to extend support
    - Collect USEFUL diagnostic info
    - Start initial diagnostic process
    - Interact with advanced diagnostics
- **Diagnostic document workflow and debug procedures**
  - Cartography - Functional map of complete system
  - Understand Diagnostic data flow - modules & protocols
- **Problem Determination (PD) automation tools**
  - Automated data collection
    - Diagnostic API's: Logs, traces, events, signals, exceptions
  - Forensic data mining => log parsing, sorting & analysis
    - Identify events leading up to problem, Isolate source(s) of problems


CONNECT
Problem Determination (PD) Tools
Diag Info Source
API's
Modifiers
View & Analysis Tools
OS
Drivers/DLL
JVM
App Svr
CONNECT App
Thread(s)
DBMS
Traces
Signals
Logs
Exceptions
Assert
Filters
Formatters
Composite Log/Trace
System Log/Trace
CONNECT LOG
Output options
Net Socket
Console
Mem Buff
File
OutputStream
SYSTEM 1
SYSTEM 2
SYSTEM 3
© CMSGateways.com


CONNECT
CMS Quality Report workflow with CONNECT/DIRECT
CONNECT
& other subsystems
Provider
CMS
Quality Report
_EHR (200+ vendors)
PM Legacy, cloud
IE
DBMS
HIE
HIE
HIE
HIE
CONNECT
FEEDBACK
File Management
XML Parser
CCD / PQRS Vetting
DBMS
© CMSGateways.com

## Contact Info

**We are developing a Field Support Toolbox for CONNECT / DIRECT**

---

**This toolbox will include a variety of Problem Resolution Tools**

---

**Please email any requirements or questions to:**

**Nick Hurd**

nickhurd@cmsgateways.com

**Thank you for participating!**