Scalable Bio-Surveillance System Architecture

Clients: Department of Defense and Civilian Public Health Authorities

Architects of Alexandria

Jichuan Chang
Owen Cheng
Madhavi Ganapathiraju
Annie Luo

May 1, 2001
Architecture for Software Systems

A. Objective:

1. BSS objective:
   The objectives of the Bio-Surveillance System is to develop and validate a scalable bio-surveillance system, the system is used to alert the DoD and Civilian Public Health authorities against suspected outbreak of diseases.

   The objectives will be achieved by development of (1) a scalable bio-surveillance architecture; (2) data mapping and merging capabilities; (3) probabilistic detection algorithms that utilize multi-source data including weather, absenteeism, web queries, and clinical data; (4) a modeling and simulation capability based on agent-based simulation methods; (5) technical and policy solutions to the protection of privacy; and (6) citywide test-beds in which prototype detection systems demonstrate our architectural and technical solutions to the problem of scalable detection systems. Project goals will be achieved by (7) management of the project teams and subcontractors.

2. Team objective:
   The objective of our team is to develop a scalable biosurveillance architecture employing an ATAM-based approach.

B. Context:

Early detection of epidemics is crucial for timely response and quarantine to prevent further spread of the disease. A system that automatically gathers pertinent information will assist in this early detection. Furthermore, such a system can be used to detect acts of bio-terrorism. Hence, DoD has initiated this project to build a Bio-surveillance System that gathers information from various health information sources such as pharmacy sales, school-absentee records, and web-based enquiries. The system interfaces with selected public health and medical systems and information processing applications. These capabilities will mitigate the impact of bio-terroristic and naturally occurring epidemic on civilian and military personnel.

C. Functional Requirements:

The BSS needs to support the following functions. The section numbers following these requirements are traceability tags pointing to the original proposal of BSS.

1. Deploy and control a set of intelligent data collection agents that obtain data from disparate, heterogeneous data sources [F.1.1.2]
2. Support different data collection methods including web-based collection, automatic data-extraction from databases, etc. [F.1.1.2]
3. The function of these data source nodes is to package the data, filter/process the data to protect privacy, protect the data via encryption, and transfer the data to the next processing unit [F.1.1.2]
4. Provide agents to convert data into canonical forms defined by existing standards such as SNOMED, LOINC and the Public Health Conceptual Data Model [F.1.1.2]
5. Provide a detection refinement or feedback mechanism where downstream agents affect the upstream collection, merging, or processing of data [F.1.1.8]
6. Provide agents to build data relationships among the various data streams such as associating observations about the same individual [F.1.1.2]
7. Implement various algorithms to detect disease outbreaks [F.1.1.2]
8. Provide user interface capabilities for alert notification and look-back queries [F.1.1.2]
9. Notify various agencies of outbreak of diseases or threat conditions reactively or proactively [F.1.1.2]
10. Implement system management capabilities to orchestrate the BSS [F.1.1.2]
11. Provide look-back query capability to trace and analyze the cause/location of outbreaks [F.1.1.2]
12. Provide dynamically adjustable access control policy [F.1.1]

D. Driving Architectural Requirements:

The BSS architecture supports the following system design aspects (the exact numbers for these requirements are given in the scenarios section of the utility tree):

1. A layered and distributed security/privacy/processing capability [F.1.1]
2. Provide a distributed agent and data interface architecture by leveraging the CoABS, DAML and TASK programs [F.1.1]
3. High availability to be achieved through redundancy of key subsystems [F.1.1.4]
4. Nearly real time performance by collection, detecting and reporting incidents in a timely way (within pre-defined threshold of latency) in order to quarantine and prevent further spread of the disease [F.1.1.10]
5. Easy addition or update of COTS/GOTS components [F.1.1]
6. Interoperate with external systems that provide data in heterogeneous formats [F.1.1.11]
7. Dynamically adjustable access control depending on risk [F.5.1.1]

E. Potential Extensions:

1. Interoperate with new civilian disease surveillance system [F.1.1.11]
2. Accommodate new data sources and support their new data formats [F.1.1.1]
3. Extensibility of interfaces between the agents and the data sources [F.1.1.1]
4. After the optional contract periods, the system may be scaled and deployed more broadly to a CONUS-wide scale and even a foreign military base [D]

F. Technical Constraints:

1. Employ relevant COTS technologies including data interchange standards (HL-7), terminology standard (SNOMED), algorithms and software for detection (Bayesian Belief Network), analysis and visualization, and security techniques [B.3]
2. Interoperation with NEDSS public health surveillance systems [F.1.1.3]
3. Comply with the DoD defense-in-depth principles to protect data security and privacy, providing hierarchical layers of protection [F.1.1.3]
4. Affordability and maintainability through the use of COTS and GOTS technologies [F.1.1]
5. Use database standards such as NEDSS and the PHCDM [F.2]

G. Business Constraints:

1. Final city-scale prototypes should be delivered in four years. [B.6]
2. The system should include transferable technology such as components for data integration, merging, detection, simulation, data visualization, look-back epidemiology and privacy [D]
3. The estimated cost of the system is $11.5 million [B.6]
4. The system should conform to the DoD standard Defense-In-Depth model [F.1.3]
5. Implementation will adhere to the DII-COE OO development standards [F.2]

Bio-surveillance System Quality Attribute Utility Tree

The Bio-surveillance System (BSS) description document isn't quite explicit about what's expected of system attributes (including performance, availability, security and modifiability addressed below). According to the document and certain assumptions, the following scenarios were generated to propagate the utility tree. Each N is a piece of fill-in-the-blank value for the client to complete, because we need realistic input from clients.

#	Quality Attributes	Characteristics	Prio.	Scenarios
1	Availability	Software Failure	(H,M)	Crucial software agent failure detected by System Management Node and recovered by backup agent within 120 sec [A-D.3]
2			(H,M)	COTS software license expiration detected, reported, and renewed, where possible, 2 weeks before expiration [A-F.4, D.5]
3		Hardware Failure	(M,H)	Data processing node failure requires incoming traffic redirected to backup node within 60 sec [A-D.3]
4		Database Failure	(H,L)	Database failures (on data processing nodes) detected within 30 sec and recovered from log file [A-D.3]
5		Survivability	(M,H)	Power/network failure at one site requires traffic redirected to a geographically separate site within 120 sec [A-D.3]
6	Modifiability	Data formats and support	(M,M)	New data collection agent can be deployed within 1 person-month [A-E.2]
7			(M,M)	Change of data format collected by data agents can be adapted within 1 person-month [A-E.3]
8			(M,H)	Change of NEDSS data model will be supported within 6 person-months. [A-F.5, F.2]
9		COTS/GOTS update & addition	(L,M)	Accommodate a updated version of COTS/GOTS within 10 person-days [A-F.4, D.5, F.1]
10			(M,M)	Integrate new detection and analysis algorithm modules within 1 person-month [A-C.7, F.1]
11		Scalability	(H,H)	The system can scale from city to state level within 60 person-months. [A-E.4]

12	Performance	Response time (Latency)	(H,M)	Report detected incidents during normal operation within 30 sec of detection [A-D.4]
13			(M,M)	Report detected incident during degraded operation within 60 sec of detection [A-D.3, D.4] (*degraded* means during transition to backup unit(s), or use of backup agents/SW of lower quality)
14			(M,H)	Lookback query latency on DBs < 5 sec [A-C.11]
15		Throughput	(L,L)	Data collection nodes record average 10 MB of data per sec [A-D.4]
16			(M,M)	Collection center DBs handles at least 10 transactions per sec [A-D.4]
17			(H,L)	Tenfold increase in detected incidents only doubles the incident report time [A-D.4]
18		Capacity	(M,L)	1 GB of data processed by data-filtering agents within a period of 120 sec [A-D.4]
19			(M,M)	Respond to data overload by instantiating additional collection or detection agents within 60 sec [A-D.3]
20			(H,H)	Data overload does not cause system to fail completely, only temporarily slow-down [A-D.3]
21	Security	Data Integrity	(H,L)	Data communication between any two nodes in the network is digitally signed by industry-level PKCS [A-D.1, F.3]
22		Data Security	(H,L)	Data transfer between any two agents is encrypted with industry-level PKCS [A-D.1, F.3]
23			(M,L)	Web-based communication is implemented with https [A-D.1, F.3]
24		Access Control	(H,L)	In the event of more than 3 consecutive login failures over the network in 60 seconds, trace level is increased and a login is disabled for 60 seconds [A-F.3, D.7]
25			(H,H)	Change in level of threat can be responded to by dynamically changing access control [A-F.3, D.7]
26			(H,L)	Connection to any machine in the system is prompted for authentication [A-D.7]
27		Privacy	(H,H)	Result of a query to the database found to be above the maximum tolerable disclosure risk level is transformed to limit disclosure potential [A-D1, D2]

At the end of this preliminary scenarios brainstorm and ranking, we ranked the importance of the four quality attributes as follows

1) Security
2) Availability
3) Modifiability, and
4) Performance.

Detailed Scenarios

In the following, we focus on scenarios of at least a Medium ranking in each criterion.

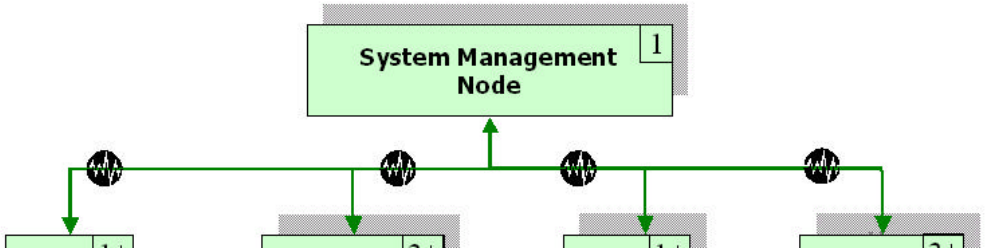
Scenario: S1 (Crucial software agent failure detected by System Management Node and recovered by backup agent within 120 sec)
Attribute: Availability
Environment: Normal operations
Stimulus: Crucial software agent failure
Response: Recovered within 120 sec

Architectural decisions	Risk	Sensitivity	Tradeoff
System Management Nodes (SMN)	X	X	
Heartbeat		X	
Backup agent (DCA, DMA, DRA, IDA, QA, AA)			

Reasoning:

- SMN provides for the monitoring and detecting of software agent failures
- Heartbeat guarantees detection within a few sec

Architecture diagram:



Heartbeat originates from the agents within nodes, and is monitored by SMN

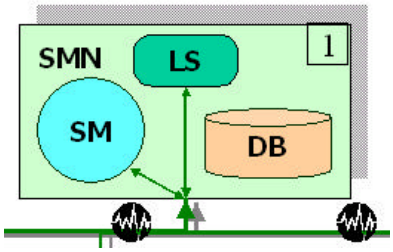
Scenario:	S2 (COTS software license expiration detected, reported, and renewed, where possible, 2 weeks before expiration)
Attribute:	Availability
Environment:	Normal operations
Stimulus:	Upcoming COTS software license expiration
Response:	Report and renew, where possible, 2 weeks before expiration

Architectural decisions	Risk	Sensitivity	Tradeoff
SMN	X	X	
License server		X	

Reasoning:

- SMN provides for the monitoring and detecting of COTS software license expiration
- License server helps centralize license management, such as renewal

Architecture diagram:



License Server resides within SMN

Scenario: S3 (Data processing node failure requires incoming traffic redirected to backup node within 60 sec)

Attribute: Availability

Environment: Normal operations

Stimulus: Data processing node failure

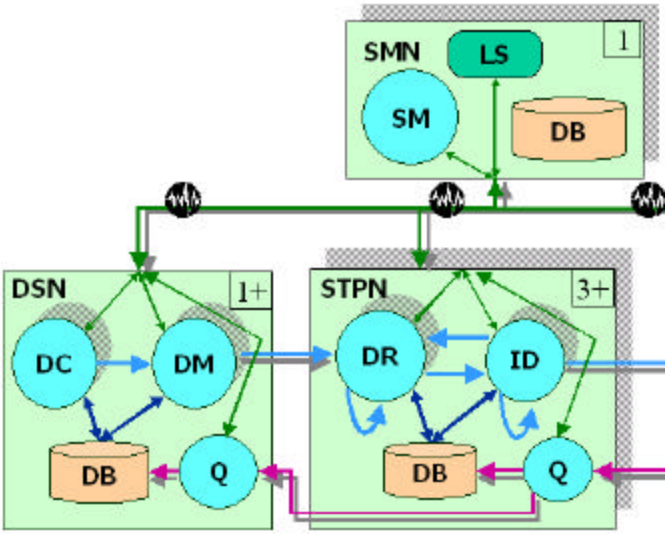
Response: Traffic redirected to backup node within 60 sec

Architectural decisions	Risk	Sensitivity	Tradeoff
Backup Data Processing Node (DPN)	X	X	
Backup communication channel	X	X	
SMN	X	X	
Heartbeat		X	
Fail-over routing		X	

Reasoning:

- Backup DPN ensures that data processing is not halted when a primary node fail
- Availability might be at risk due to the lack of backup communication channel
- SMN ensures detection of node failure and initiates traffic redirection
- Heartbeat guarantees detection within a few sec
- Fail-over routing ensures instant traffic routing upon detection of failure; it decreases the risk of SMN

Architecture diagram:



Fail-over routing resides on the network to route traffic to backup SMN

Scenario: S5 (Power/network failure at one site requires traffic redirected to a geographically separate site within 120 sec)

Attribute: Availability

Environment: Normal operations

Stimulus: Power or network failure at a certain site

Response: Traffic redirected to a geographically separate site within 120 sec

Architectural decisions	Risk	Sensitivity	Tradeoff
Geographically separate backup site	X	X	
Backup data channel	X	X	
Geographically separate peer SMN	X	X	
Heartbeat		X	

Reasoning:

- Geographically separate backup site ensures that system is not paralyzed by power failure or massive network failure in one site
- Availability might be at risk due to the lack of backup data channel
- Geographically separate SMN ensures detection of site failure and initiates traffic redirection
- Heartbeat guarantees detection within a few sec

Architecture diagram:

Refer to the architectural diagram in S3

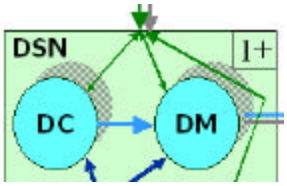
Scenario: S6 (New data collection agent can be deployed within 1 person-month)
Attribute: Modifiability
Environment: Normal operations
Stimulus: New data collection agent required
Response: DCA deployed within 1 person-month

Architectural decisions	Risk	Sensitivity	Tradeoff
Loose coupling of agent functionality	X	X	
Standard data exchange protocol	X	X	

Reasoning:

- Loose coupling enables addition of new agent without having to modify other agents
- Standard data exchange protocol speeds up new agent deployment

Architecture diagram:



The only interaction between data agents is data-flow, indicating loose-coupling between them
Data Mapping agent enforces standard data exchange protocol

Scenario: S7 (Change of data format collected by data agents can be adapted within 1 person-month)

Attribute: Modifiability

Environment: Normal operations

Stimulus: Data format change

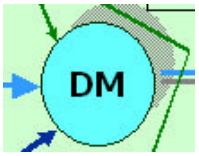
Response: Affected data agents are adapted within 1 person-month

Architectural decisions	Risk	Sensitivity	Tradeoff
Data mapping agent (DMA)	X	X	
Standard data exchange protocol	X	X	

Reasoning:

- DMA encapsulates different data formats from internal system
- Standard data exchange protocol speeds up new agent deployment

Architecture diagram:



Data Mapping agent enforces standard data exchange protocol

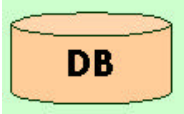
Scenario: S8 (Change of NEDSS data model will be supported within 6 person-months)
Attribute: Modifiability
Environment: Normal operations
Stimulus: NEDSS data model change
Response: Supported by COTS database within 6 person-months

Architectural decisions	Risk	Sensitivity	Tradeoff
COTS database	X	X	

Reasoning:

- COTS database supports the change of model

Architecture diagram:



Note that DB implies the existence of DBMS

Scenario: S10 (Integrate new detection and analysis algorithm modules within 1 person-month)

Attribute: Modifiability

Environment: Normal operations

Stimulus: New modules with updated detection and analysis algorithms

Response: Integrated within 1 person-month

Architectural decisions	Risk	Sensitivity	Tradeoff
Loose coupling of agent functionality	X	X	

Reasoning:

- Loose coupling enables modification of detection agents without having to modify other agents

Architecture diagram:

Refer to the architectural diagram in S6

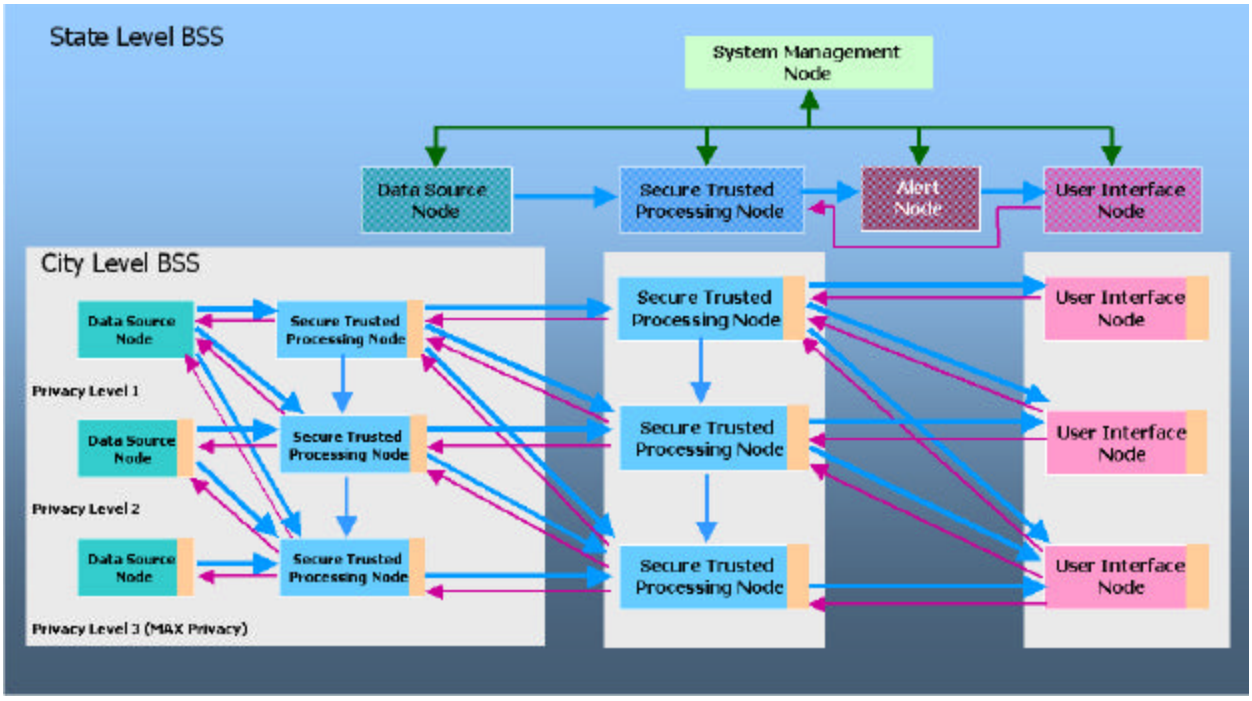
Scenario: S11 (The system can scale from city to state level within 60 person-months)
Attribute: Modifiability
Environment: Normal operations
Stimulus: Need to scale the system to state level
Response: Completed within 60 person-months

Architectural decisions	Risk	Sensitivity	Tradeoff
Repeatable architectural pattern	X	X	
Hierarchical structure		X	

Reasoning:

- Same architectural pattern for each level, with connectors from one level to the next
- Hierarchical structure allows treatment of lower level structure as one component in the higher level

Architecture diagram:



Secure Trusted Processing Nodes at one level become the Data Source Node for the next higher level

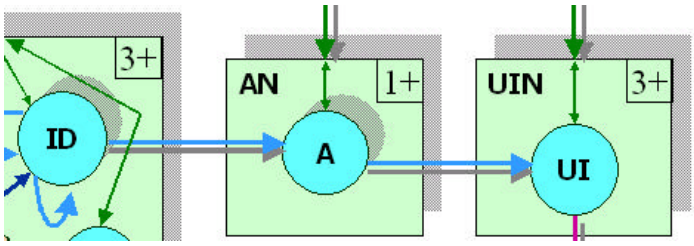
Scenario:	S12 (Report detected incidents during normal operation within 30 sec of detection)
Attribute:	Performance
Environment:	Normal operations (i.e. nodes work properly, but communication link may or may not)
Stimulus:	Incident detected
Response:	Reported within 30 seconds

Architectural decisions	Risk	Sensitivity	Tradeoff
Redundant communication channels	X	X	

Reasoning:

- Redundant communication channels ensure that the detected incident gets reported to the Alert Nodes (AN) and alert is transmitted to the User Interface Node (UIN)

Architecture diagram:



Redundant communication channel ensures alert gets reported to AN and transmitted to UIN

Scenario: S13 (Report detected incident during degraded operation within 60 sec of detection)

Attribute: Performance

Environment: Degraded operations

Stimulus: Incident detected

Response: Reported within 60 seconds

Architectural decisions	Risk	Sensitivity	Tradeoff
Backup alert nodes (AN)	X	X	
Backup alert agents (AA)		X	
Redundant communication channels	X	X	

Reasoning:

- Backup AN ensures that detected incidents get processed when the primary node fails
- Backup AA ensures that detected incidents get processed when the primary agent in a node fails
- Redundant communication channels ensure that the detected incident gets reported to the AN and alert is transmitted to UIN

Architecture diagram:

Refer to the architectural diagram in S12
Backup AN, AA and redundant communication channel ensures alert gets reported to AN and transmitted to UIN

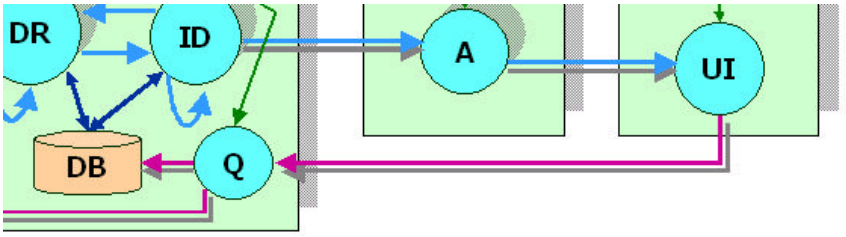
Scenario: S14 (Lookback query latency on DBs < 5 seconds)
Attribute: Performance
Environment: Normal operations
Stimulus: Lookback query initiated
Response: Partial response (part of total result set) returned within 5 seconds

Architectural decisions	Risk	Sensitivity	Tradeoff
Intelligent User Interface Agents (UIA)		X	
Backup communication channel	X	X	
Query agent	X	X	

Reasoning:

- Intelligent UIA delivers information as it comes without waiting for the complete response
- Backup data channel guarantees transmission of query results without delay
- Database and query agent support for incremental return of query result reduces delay

Architecture diagram:



The flow of query result is in fact a pipe-like data flow, which inherently supports incremental transfer of results

Scenario: S16 (Collection center DBs handles at least 10 transactions per second)
Attribute: Performance
Environment: Normal operations
Stimulus: Incoming transaction requests
Response: Process 10 requests within 1 second

Architectural decisions	Risk	Sensitivity	Tradeoff
Efficient DBMS	X	X	

Reasoning:

- Efficient DBMS provides sufficient processing capability

Architecture diagram:

Refer to the architectural diagram in S8

Scenario: S19 (Respond to data overload by instantiating additional collection or detection agents within 60 seconds)

Attribute: Performance

Environment: Normal operations

Stimulus: Data overload

Response: Additional collection or detection agents instantiated within 60 seconds

Architectural decisions	Risk	Sensitivity	Tradeoff
SMN	X	X	
Agent self-monitoring capability		X	

Reasoning:

- SMN monitors for agent data overload and instantiates additional agents
- Agent self-monitoring capability detects data overload

Architecture diagram:

Refer to the architectural diagram in S2
Agent self-monitoring capability is not shown in the diagram

Scenario: S20 (Data overload does not cause system to fail completely, only temporarily slow-down)

Attribute: Performance

Environment: Normal operations

Stimulus: Data overload

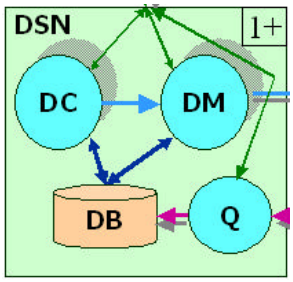
Response: System continues operation

Architectural decisions	Risk	Sensitivity	Tradeoff
Redundant agents	X	X	
Efficient DBMS	X	X	

Reasoning:

- Redundant agents alleviate data overload
- Efficient DBMS provides sufficient processing capability

Architecture diagram:



Redundant Data Collection and Data Mapping Agents alleviate data overload

Scenario: S25 (Change in level of threat can be responded to by dynamically changing access control)

Attribute: Security

Environment: Normal operations

Stimulus: Abnormally high activity of network access

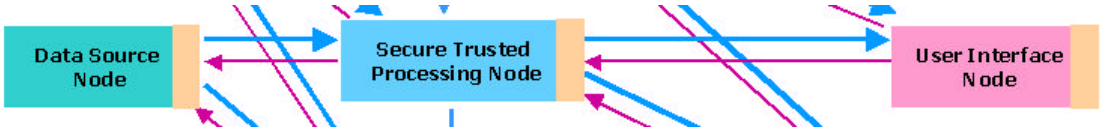
Response: Escalation of access restriction

Architectural decisions	Risk	Sensitivity	Tradeoff
Access control module	X	X	

Reasoning:

- Dynamic access control module monitors the network activities and escalates access restriction

Architecture diagram:



Dynamic access control shields the nodes from the network traffic

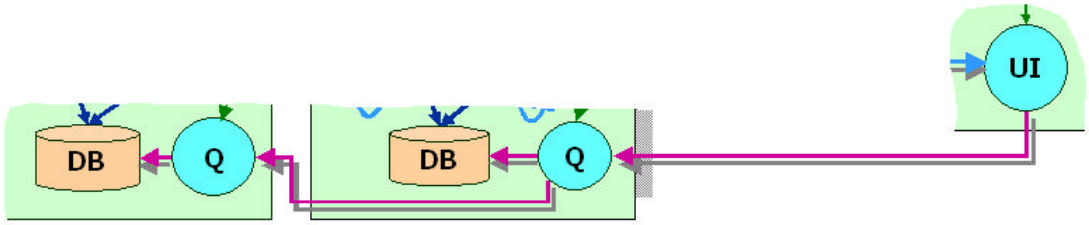
Scenario:	S27 (Result of a query to the database found to be above the maximum tolerable disclosure risk level is transformed to limit disclosure potential)
Attribute:	Security
Environment:	Normal operations
Stimulus:	Result of a query to the database found to be above the maximum tolerable disclosure risk level
Response:	The result is transformed to limit disclosure potential

Architectural decisions	Risk	Sensitivity	Tradeoff
Query agent	X	X	

Reasoning:

- Query agent on the SPTN assess the disclosure risk level of query result and transforms the query result

Architecture diagram:



The query agent assesses the disclosure risk level according to its privacy layer

Sensitivity Point Analysis

This section lists the sensitivity points identified from this ATAM exercise. For the reasoning, please refer to the Detailed Scenario tables above.

SP1. SMN
SP2. Heartbeat
SP3. License Server
SP4. Backup DPN
SP5. Backup data channel
SP6. Backup/redundant communication channel
SP7. Fail-over routing
SP8. Geographically separate site
SP9. Geographically separate SMN
SP10. Loose coupling
SP11. Standard data exchange protocol
SP12. DMA
SP13. COTS Database
SP14. Repeatable architectural pattern
SP15. Hierarchical structure
SP16. Backup AN
SP17. Backup AA
SP18. Intelligent UIA
SP19. QA
SP20. Efficient DBMS
SP21. Agent self-monitoring
SP22. Redundant/backup agent
SP23. Access control module

Risk Analysis

R1. SMN:
Without SMN, agent failures can neither be detected nor recovered, and thus, the availability of the system is compromised. In addition, software license expiration will also go undetected. SMN is also needed to respond to node failure, and agent data overload.

R2. Backup DPN:
Without backup DPN, data processing will halt if the primary node fails, and no incident detection will occur, thus compromising system availability.

R3. Backup communication channel:
Availability might be at risk due to the lack of backup communication channel: in the case that the primary channel fails, data transmission will be halted, thereby halting the system.

R4. Geographically separate site:
Without a geographically separate site, the system will be paralyzed by network or power failure at one site.

R5. Geographically separate peer SMN:
Site failure will not be detected without geographically separate, peer SMN, because the primary SMN is likely to have failed with the primary site, which leads to R4.

R6. Loose coupling:
Changes to one agent in the absence of loose coupling might affect other agents, thus decreasing modifiability of the overall system.

R7. Standard data exchange protocol:
Without a standard protocol, changes to one data source format could require otherwise unnecessary modifications of the system downstream.

R8. DMA:
Different data format cannot be encapsulated from the system downstream from the data sources without DMAs.

R9. COTS Database:
Without COTS database, data management capabilities will have to be separately implemented by the system developer, thus increasing the development time, and also compromising modifiability.

R10. Repeatable architectural pattern:
Without a repeatable architectural pattern, the system cannot scale easily from one level to the next geographical level.

R11. Redundant communication channel:
Without redundant communication channel, detected incidents may not reach the UIN in time due to network delay or failure.

R12. Backup AN:
Without backup AN, notification of detected incidents will not occur in time if the primary AN fails.

R13. QA:
Absence of query agent might hinder incremental return of query result, which impacts performance of the look-back capability. In addition, its absence prevents the enforcement of privacy policy on information disclosure.

R14. Efficient DBMS:
Absence of efficient DBMS might affect the speed of processing data, delaying incident detection to beyond acceptable level, and also the performance of look-back query might be affected.

R15. Redundant/backup agent:
Without redundant/backup agents, failure of single agent might affect the functionality of the corresponding part of the system.

R16. Access control module:
Absence of access control module prevents detection of abnormal network access and could compromise system security.

Trade-Off Analysis

T1. SMN:
Although it affects performance and availability, it is not a trade-off because it improves both attributes.

T2. Backup data channel:
Although it affects performance and availability, it is not a trade-off because it improves both attributes.

T3. QA:
It is a trade-off point between performance and security because detection of sensitive information disclosure and the transformation of query result might affect the performance, while the requirement for performance might compromise privacy enforcement.

Key Decisions and Analysis of the Architectural Design

We now discuss the key decisions for our architectural design derived from analysis of the scenarios. In the architectural figures that follow, the components are the nodes, agents, servers and databases; the connectors are the various flows, including the heartbeat.

Figure 1 shows the context diagram for the Bio-Surveillance System.

Figure 1: System Context Diagram

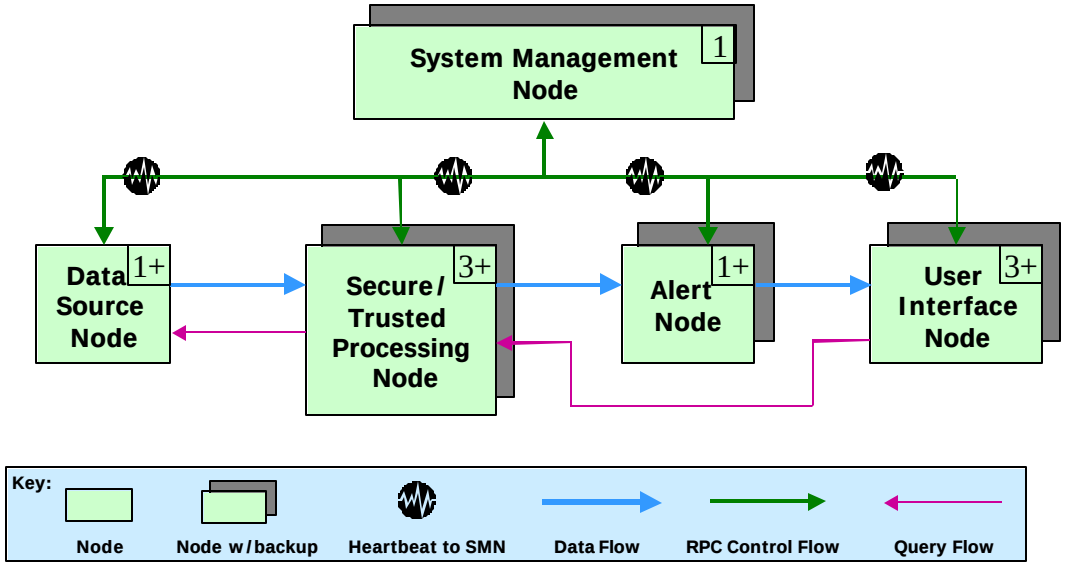


Figure 2: Top-level modified Pipe-and-Filter architecture of BSS

The proposed architecture for the Bio-Surveillance System consists of Data Source Nodes (DSN), System Management Node (SMN), Secure/Trusted Processing Nodes (STPN), Alert

Nodes (AN) and User Interface Nodes (UIN), which interact as described below. The SMN controls the other nodes through Remote Procedure Calls. These nodes, including the SMN, are connected in a modified pipe-and-filter architectural style as shown in Figure 2. Figure 2 illustrates the architecture for one unit of data collection, which typically covers the area of a city. In each unit, there may be one or more DSNs, at least three layers of STPNs to ensure privacy and security, one or more ANs, one SMN, and three or more UINs, also for each privacy layer. The UINs can perform look-back query using database connection to the STPN, which can in turn query the DSN.

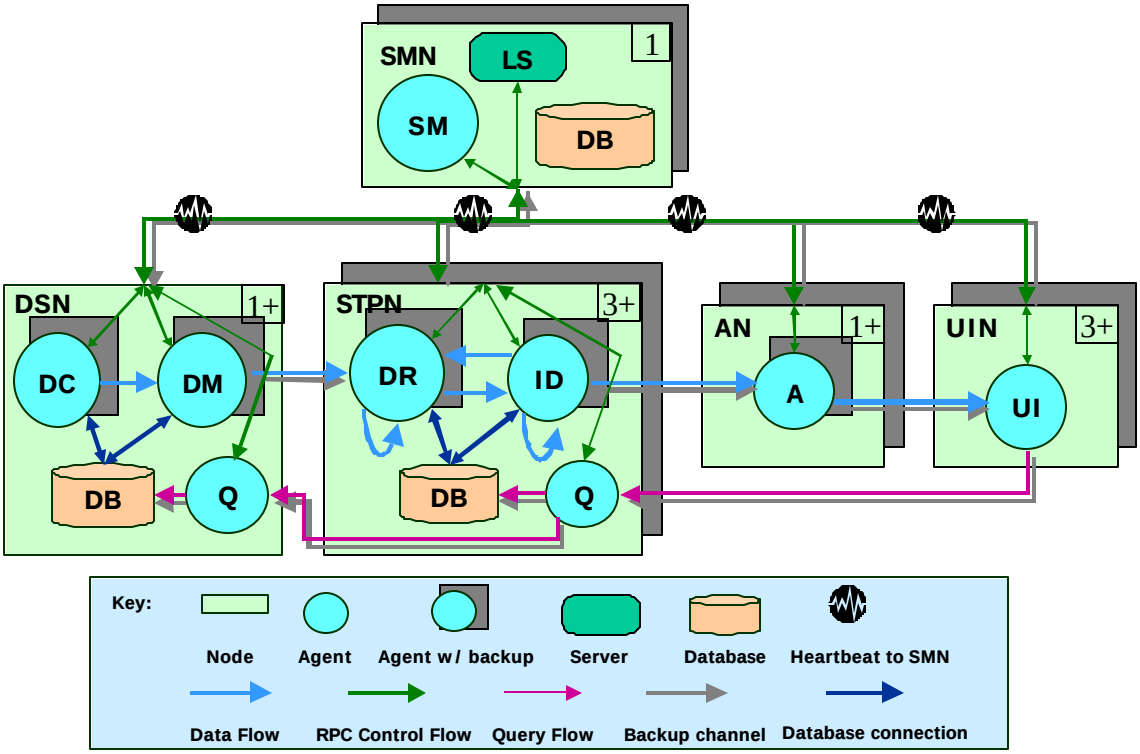


Figure 3: Decomposition of nodes in the BSS architecture

Figure **3** shows a pattern of connectivity between the agents/nodes. Specifically, the connections from the DRA to the IDA and back are not between the same instances of the two; this is emphasized by the use of two separate arrows as opposed to a bi-directional arrow. In Figure **3**, each node of Figure 2 is further decomposed as follows:

- The System Management Agent (SMA) in SMN performs coordination activities between agents residing in all other nodes. It monitors the heartbeats (periodic liveness signals sent by agents to SMN), thus ensuring that the failure of any node is detected within a short period of time. It also responds to the alerts of data overload from all the data collection and detection agents by spawning the backup or redundant agents as necessary. To ensure quicker response to data overload situations, agents could be equipped with self-monitoring capabilities, but this alternative was not chosen to avoid complicating agent implementation.

- SMN also employs the License Server (LS) to detect the expiration of the COTS and GOTS licenses in every node and renews them where possible or notifies system administrator otherwise.
- The data sources, which are outside of BSS, could be disparate and heterogeneous. Each data source site has a DSN deployed on it.
- A DSN consists of Data Collection Agents (DCA), Data Mapping Agents (DMA), Query Agents (QA) and a database. The DCA collects data from its data source and sends it to the DMA, which in turn filters the data and processes it to protect privacy, encrypts it and transfers it to the Data Relationship Agents (DRA) in the STPN.
- The STPN consists of DRAs, Incident Detection Agents (IDA), QAs and a database. The DRA builds data relationships among the various data streams, converts data into canonical forms defined by specific standards and sends it to the IDA, which implements various algorithms to detect disease outbreaks and transfers the information to the Alert Agent (AA) in the AN.
- The AN employs AA to notify various agencies of outbreak of diseases or threat conditions reactively or proactively, the alert is sent to the User Interface Agent (UIA) in the UIN, which provides user interface capabilities for alert notification. To ensure that detected incidents get processed in the event of alert agent or node failure, backup AN and AA are designed into the system.
- The UIA in the UIN provides interface to notify users of incidents detected by the system, and also allows the look-back query capacity, which may be automatic or be assisted by authenticated users, to collect additional data to clarify the nature of a suspicious case or pattern of cases.
- QAs, though not prescribed in the client requirement, were deemed important in order to increase query performance and assess disclosure risk level for look back queries.
- An efficient Database Management System (DB) in each node provides the persistent storage in the DSN, STPN and SMN separately. The DB in DSN and STPN is open to be queried from the UIN through their corresponding QAs.
- Every node except the DSN is backed up to ensure availability and reliable functionality.
- In order to ensure system availability, backups are provided for crucial data and communication channels.
- Data flows from each data source through its corresponding DCA. The DCA writes the data to the database and also sends it to DMA. The DRA accepts data from various DMAs, processes it iteratively, and, when complete, passes the data directly to IDA and writes it to database. DRA can also accept data from IDA in order to provide traceability information related to any incident. The AA in AN gets incident data from IDA and passes it to the UIA in a UIN of an appropriate privacy level. The data flow channels between nodes are backed up to ensure availability. Redundant data channels are provided between nodes to cope with network failures external to the system.
- SMN orchestrates other nodes and agents via Remote Procedure Call (RPC) control flow.
- Agents send heartbeats to SMN via RPC mechanism.
- The look-back query ability is executed via query flow.

In order to provide adequate privacy and security, the overall BSS system architecture will follow the Defense-in-Depth (DID) principles. Figure 4 focuses on the privacy and security issues, and hence omits certain details from the previous figures. It shows the hierarchical layers of privacy protection throughout the system and exposes the access control modules within the processing nodes.

- Each of the DSNs, STPNs, and UINs can belong to any one of the privacy levels. The data flow between these nodes is restricted to flow from a lower privacy level to a higher privacy level, but not the other way around. Likewise, the query flow between these nodes is restricted to originate from a higher privacy level to a lower privacy level, but not the other way around.
- The choice of which privacy level a DSN belongs to is a decision of the corresponding data provider. For example, veterinarian records are most likely to be in the lowest privacy level, whereas human medical records are likely to be in the highest privacy level.
- A dynamic access control module shields each of the nodes, except the DSNs in the lowest privacy level.
- The dynamic access control modules monitors network activities, assesses risk level, and escalates access restriction when the risk level increases. The access restriction can also be adjusted manually by authorized users.
- The access control modules on the UINs control access by users.
- Data can flow from an STPN at a lower privacy level to an STPN at a higher privacy level. This can occur as data flow from IDAs to IDAs, DRAs to DRAs, DRAs to IDAs, or IDAs to DRAs.

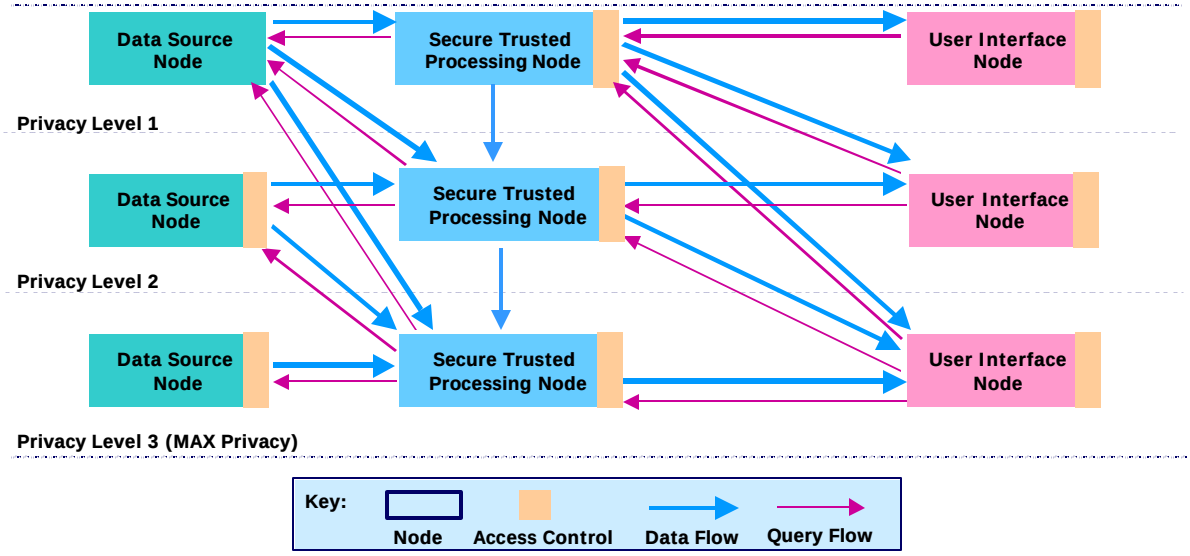


Figure 4: Explicit representation of privacy protection levels and access control within processing nodes

Potential Extensions

- Scale the system from city level to state level:
  The current architecture uses repeatable architectural pattern for each geographical level, with new data and query flow connector instances from one level to the next. No additional types of component or connector are needed. This property of the architecture supports scalability without having to modify or redesign the entire architecture. Hierarchical structure allows treatment of lower level structure as one component in the higher level.

- Accommodate new data sources:
  To support new data sources, we need to add new DCAs. Because of loose coupling and standard data exchange protocol, addition of new agents can be realized without affecting the rest of the system.
- Change of data formats in data sources:
  Due to loose coupling, change of data formats in the data source affects only the DCA.
- Update of COTS/GOTS components:
  The BSS architecture depends primarily on data flows between agents and is thus independent of COTS/GOTS interfaces. Therefore, maintenance between versions remains tractable. For the data processing agents, it is possible that their input/output data formats are modified, but this can be accommodated by data wrappers. If COTS/GOTS components do not support heartbeat, separate monitors can be implemented to accomplish the task. Since SMAs perform functions quite specific to BSS, they are developed in-house and thus undesirable interface changes should not occur.
- Change of data model
  COTS database provides support for data model changes, which encapsulates the changes from the rest of the system.

Evaluation of the ATAM Experience

The process of mapping requirements to design is intractable without a systematic approach. Generating the architectural designs from the BSS requirements document was an arduous process even with the help of ATAM. Although ATAM provides a systematic approach, it was designed to evaluate the appropriateness of an architectural design in the presence of technical and business constraints, but not to build the architectural design itself. Hence the preliminary architectural design came mostly from our brainstorming, and did not benefit much from using ATAM.

However, once the initial design took shape, ATAM could be used iteratively to improve the architecture. Without ATAM, it is very difficult to determine early on how well the architectural design satisfies the requirements. ATAM helped to highlight important quality attributes, which allowed us to discover and weigh design alternatives, and provided a rationale for decision-making. The exercise of generating the scenarios and prioritizing them contributed to our understanding and helped us in sensitizing the architecture to the relevant quality attributes.

Expanding the important scenarios drove us to make finer analyses and incrementally enhance our design. ATAM also provided a template for organizing and documenting scenario details and the corresponding architectural design decisions, making the rationale explicit. During the process of identifying risks and sensitivity points, missing architectural elements were discovered. The process of finding tradeoff points brought our attention to one quality attribute that was not considered earlier (i.e. Affordability), by which time it was too late to be incorporated. This was due mainly to the lack of participation from the customers. If customer involvement were available and active, finding the tradeoff points would have assured us that no other tradeoff points existed in our design.