

**PMR Insight Collective Knowledge (PICK)
Class-responsibility-collaboration Model**

Version 1.1

3/2/2020

Document Control

Approval

The Guidance Team and the customer shall approve this document.

Document Change Control

Initial Release:	1.1
Current Release:	1.1
Indicator of Last Page in Document:	[END]
Date of Last Review:	03/02/2020
Date of Next Review:	03/02/2020
Target Date for Next Update:	03/09/2020

Distribution List

This following list of people shall receive a copy of this document every time a new version of this document becomes available:

Guidance Team Members:

Dr. Ann Gates, Dr. Steve Roach, Mr. Jake Lasley

Clients:

Mr. Anthony DesArmier, Mr. Angel Villalpando, Mr. Jorge Garcia, Mr. David Rayner, Mr. Valentin Becerra, Mr. Mario Delgado

Software Team Members:

Mr. Anthony DesArmier, Mr. Angel Villalpando, Mr. Jorge Garcia, Mr. David Rayner, Mr. Valentin Becerra, Mr. Mario Delgado

Change Summary

The following table details changes made between versions of this document

Version	Date	Modifier	Description
0.1	2/14/2020	David Rayner	Added template Added Settings, Node, Graph, ProjectMerge, History, UIChangeConfig, UIVectorConfig, UI
0.1	2/14/2020	Angel Villalpando	Added UiDirectoryConfig, UiEventConfig, UiFilterConfig, UiVectorDBLead, LogFile, EnforcementActionReport, Validator, ExportTable, ExportGraph
0.2	2/16/2020	Jorge Garcia	Added UiVectorConfig, formatting
0.2	2/16/2020	Valentin Becerra	Node, Log Entry, Graph, Vector
0.3	2/17/2020	Mario Delgado	Edited document, made changes to UiVectorConfig
1.0	2/17/2020	Anthony DesArmier	Verified UiDirectoryConfig, UiEventConfig, UiFilterConfig, UiVectorDBLead, LogFile, EnforcementActionReport, Validator, ExportTable, ExportGraph

CRC	Keikaku 企画	Date 3/2/2020 9:26 PM	Page 1
-----	------------	--------------------------	-----------

Class-responsibility-collaboration Model

1.1	3/2/2020	Software Team	Added/linked class collaborators responsibilities. Added Sync, QObject, Dictionary, and SplunkManager CRCs
-----	----------	---------------	---

Class: UiChangeConfig

Description: Commit window handles committing of log entry and vector changes.

Responsibilities:

1. Display vector changes. (History 3)
2. Display node changes. (History 3)
3. Display relationship changes. (History 3)
4. Display log entry changes. (History 3)
5. Commit button triggers save action. (History 2)

Collaborators:

- History

Class: UiDirectoryConfig

Description: Handles the setting of file or file path directories.

Responsibilities:

1. Collects user provided directory path for root ingestion. (Settings 4)
2. Collects red team directory path. (Settings 3)
3. Collects blue team directory path. (Settings 3)
4. Collects white team directory path. (Settings 3)

Collaborators:

- Settings

Class: UiEventConfig

Description: Handles the setting of project event information.

Responsibilities:

1. Collects event name. (Event 1)
2. Collects event description. (Event 2)
3. Collects event start timestamp. (Event 3)
4. Collects event end timestamp. (Event 4)

Collaborators:

- Event

Class: UiExportConfig

Description: The export window which handles exporting vector and entry data to an external file.

Responsibilities:

1. Allows user to select directory.
2. Allows user to select file format.
3. Allows user to name file to export.
4. Creates an image file of vector graph. (ExportGraph 1)
5. Creates a csv file of vector table. (ExportTable 6)

Collaborators:

- ExportTable
- ExportGraph

Class: UiFilterConfig

Description: Collects, handles, and applies the search criteria to the system generated log entries.

Responsibilities:

1. Allows user to filter log file, log entry, relationship and node table.
2. Collects keywords to search for in generated log entries.
3. Collects the start/end timestamp search-criteria to apply to generated log entries.
4. Collects the “creator” (red, white, or blue) search criteria for generated log entries.
5. Collects the “event type” (red, white, or blue) search-criteria for generated log files.

Collaborators:

Class: UiTeamConfig

Description: The window which handles the system role and connections to the host when running as a client.

Responsibilities:

1. Allows user to define if they are a lead or not.
2. If lead is checked, displays the host’s IP address. (Settings 1; Settings 2)
3. If lead is not checked, displays the lead’s IP address. (Settings 1; Settings 3)
4. Displays the number of connections to the lead’s IP address. (Settings 3, Settings 7)

Collaborators:

- Settings

Class: UiVectorConfig

Description: The vector window which handles the adding, editing, and deleting of vectors and their descriptions.

Responsibilities:

1. Allows user to add a new vector. (Dictionary 1)
2. Allows user to delete a vector. (Dictionary 2)
3. Allows user to edit existing vector attributes. (Dictionary 3; Vector 1, 2)
4. Displays vector name and description. (Dictionary 3; Vector 1, 2)

Collaborators:

- Vector
- Dictionary

Class: UiRelationshipConfig

Description: Handles the relationship table for the active vector.

Responsibilities:

1. Display relationship id, parent id, child id, and label. (Relationship 1, 2, 4, 5)
2. Collects parent id. (Relationship 1)
3. Collects child id. (Relationship 2)
4. Allows user to enter label description. (Relationship 4)
5. Allows user to add new relationship entry. (Dictionary 1)
6. Allows user to remove a relationship entry. (Dictionary 2)
7. Allows user to edit relationship entry. (Dictionary 3; Relationship 1, 2, 4, 5)

Collaborators:

- Relationship
- Dictionary

CRC	Keikaku 企画	Date 3/2/2020 9:26 PM	Page 4
-----	------------	--------------------------	-----------

Class: UiVectorDBAnalyst

Description: The analyst vector DB window which handles the pulling and pushing of vector changes between the analyst and host systems.

Responsibilities:

1. Display vector lead's pull table. (Dictionary 3; Vector 1, 2; Graph 5)
2. Display vector client's push table. (Dictionary 3; Vector 1, 2; Graph 5)
3. Allows user to pull lead's table. (Sync 2; Dictionary 3; Vector 1, 2; ProjectMerge 2)
4. Allows user to push client's table. (Sync 1; Dictionary 3; Vector 1, 2; ProjectMerge 2)

Collaborators:

- ProjectMerge
- Vector
- Graph
- Sync

Class: UiVectorDBLead

Description: The lead vector DB window which handles the accepting of queued analyst push requests.

Responsibilities:

1. Allows user to accept push requests to the database. (ProjectMerge 2)
2. Allows user to decline push requests to the database.
3. Display information to be pushed to the database. (Vector 1, 2; Graph 5; Sync 3)

Collaborators:

- ProjectMerge
- Vector
- Graph
- Sync

Class: UI

Description: The main window which serves as entry point to the application and provides the bulk of the system's interface.

Responsibilities:

1. Launching all UI dialogs.
2. Open appropriate UiVectorDBLead or UiVectorDBAnalyst. (Settings 1)
3. Display vector table. (Dictionary 3; Vector 1, 2)
4. Allows user to add, edit and delete nodes. (Dictionary 1, 2, 3; Node 1)
5. Allows user to select node property visibilities. (Settings 8)
6. Allows user to set node visibility. (Node 4)
7. Display vector graphical view. (Graph 5)
8. Allows user to zoom in/out on the graphical view.
9. Allows user to change the timeline orientation.
10. Allows user to change the timeline interval.
11. Allows user to select between different vectors. (Dictionary 3)
12. Displays log entry table. (Dictionary 3; LogEntry 2, 3, 4)
13. Allows user the ability to edit log entries. (Dictionary 3, LogEntry 2, 3, 4)
14. Displays log file table. (Dictionary 3, LogFile 1, 2, 3, 5)
15. Displays Enforcement Action Report table. (EnforcementActionReport 1, 2)
16. Allows user to select an Enforcement Action Report to display. (LogFile 6)
17. Allows user to revalidate log files. (Validator 3)
18. Stores the vector dictionary.
19. Keeps track of the active vector. (ActiveVector 1, 2)

CRC	Keikaku 企画	Date 3/2/2020 9:26 PM	Page 5
-----	------------	--------------------------	-----------

Collaborators:

- Node
- Vector
- Graph
- Dictionary
- EnforcementActionReport
- Validator
- LogFile
- Settings
- ActiveVector

Class: History

Description: Keeps track of the log entries, nodes, and vectors and relationships history of changes.

Responsibilities:

1. Allows user to undo commit action. (Node 1-5; LogEntry 1-3; Relationship 1-4; Vector 1-4)
2. Stores user actions in sequence.
3. Provides commit information. (Node 1-5; LogEntry 1-3; Relationship 1-4; Vector 1-4)

Collaborators:

- Node
- LogEntry
- Relationship
- Vector

Class: Event

Description: Maintains the information and attributes that describe the specific event in a given analysis/session.

Responsibilities:

1. Stores the user-provided event name.
2. Stores the user-provided event description.
3. Stores the user-provided start timestamp.
4. Stores the user-provided end timestamp.

Collaborators:

Class: LogFile

Description: The log file class will read the user-provided directory path and import the contents of the file.

Responsibilities:

1. Keeps track of cleansing status. (Validator 1, 2)
2. Keeps track of validation status. (Validator 4)
3. Keeps track of ingestion status.
4. Creates copy of cleansed file.
5. Stores the log file name. (Settings 4, 5)
6. Generates enforcement action report upon error detection.

Collaborators:

- Validator
- EnforcementActionReport
- Settings

CRC	Keikaku 企画	Date 3/2/2020 9:26 PM	Page 6
-----	------------	--------------------------	-----------

Class: LogEntry

Description: The log entry class will store the indexed entries in a log file.

Responsibilities:

1. Generates unique id.
2. Stores entry timestamp.
3. Stores entry description.
4. Stores a reference to the log file. (LogFile 5)

Collaborators:

- LogFile

Class: Graph

Description: Keeps track of nodes, relationships, their relative positions, and their properties.

Responsibilities:

1. Stores node positions. (Node 5)
2. Stores relationship positions. (Relationship 3)
3. Stores graph orientation.
4. Store timeline intervals.
5. Renders nodes and relationships according to graph orientation and timeline interval. (Node 1-5; Relationship 1-5)

Collaborators:

- Node
- Relationship

Class: Node

Description: Responsible for the storage of log entry properties.

Responsibilities:

1. Stores node name.
2. Stores node Id.
3. Stores node timestamp.
4. Stores node description.
5. Stores node log entry reference.
6. Stores node Log Creator.
7. Stores node Event Type.
8. Stores node Icon Type.
9. Stores node source file.
10. Generates a unique node id.
11. Keeps track of node visibility.
12. Keeps track of node coordinate position.

Collaborators:

Class: Vector

Description: Responsible for storing the data of nodes and relationships that compose a vector.

Responsibilities:

1. Stores vector name.
2. Stores vector description.
3. Adding new nodes to dictionary. (Dictionary 1)
4. Removing nodes from dictionary. (Dictionary 2)
5. Retrieves nodes form dictionary. (Dictionary 3)
6. Add new relationships to dictionary. (Dictionary 1)
7. Remove relationships from entry. (Dictionary 2)

CRC	Keikaku 企画	Date 3/2/2020 9:26 PM	Page 7
-----	------------	--------------------------	-----------

- Retrieves relationships from dictionary. (Dictionary 3)

Collaborators:

- Dictionary
- UI

Class: Relationship

Description: Stores the child and parent node IDs that constitute a given relationship in the system.

Responsibilities:

- Stores the node identification number for a given parent node.
- Stores the node identification number for a given child node.
- Stores relationship line coordinates.
- Stores the relationship label.
- Generates a relationship ID for a parent and a child node.

Collaborators:

Class: EnforcementActionReport

Description: Report that outlines various errors found in validation and solicits the user for an enforcement action.

Responsibilities:

- Stores the errors detected by Validator.
- Determines what action must be performed depending on the error detected.

Collaborators:

Class: ProjectMerge

Description: Handles the merging of analyst and lead changes.

- Manages merge of data for analyst on pull of lead changes. (History 3)
- Manages merge of data for lead on push of analyst changes. (History 3)

Collaborators:

- History

Class: Validator

Description: Responsible for validating and cleansing the content of the user-provided file(s).

Responsibilities:

- Cleanses TMUX log files for unwanted characters.
- Cleanses log files such as removing empty lines/rows.
- Determines what errors are found in the user-provided files.

Collaborators:

Class: Settings

Description: Responsible for housing global variables of a given event/session.

Responsibility:

- Stores the lead status check.
- Stores the host machine IP address.
- Stores the lead machine IP address.
- Stores the Red, Blue, and White team directories.
- Stores the root directory.
- Stores the icon directory.
- Keep track of the number of connections to lead.
- Keeps track of node property visibility flags.

Collaborators:

CRC	Keikaku 企画	Date 3/2/2020 9:26 PM	Page 8
-----	------------	--------------------------	-----------

Class: ExportTable

Description: Responsible for the formatting of the log entry tables that compose the vectors of a given event into an exportable file.

Responsibilities:

1. Compiles the node and relationship table into a single, exportable file. (Vector 1, 2; Node 1-9; Relationship 1, 2, 4)

Collaborators:

- Vector
- Relationship
- Node

Class: ExportGraph

Description: Responsible for the formatting of the final graph with its corresponding vectors into an exportable image file.

1. Renders an exportable image file of the Graph. (Graph 5)

Collaborators:

- Graph

Class: Dictionary

Description: Keeps track of all generic objects with a unique id.

Responsibilities:

1. Adds an entry to the dictionary. (QObject 1)
2. Removes an entry from the dictionary. (QObject 1)
3. Retrieves an entry from the dictionary. (QObject 1)

Collaborators:

- QObject

Class: QObject

Description: To detect and notify in the event a vector is changed.

Responsibilities:

1. Emit pyQT signals.

Collaborators:

Class: ActiveVector

Description: Keeps track of the vector currently selected.

Responsibilities:

1. Keeps track of the active vector.
2. Keeps track of the active vector id.

Collaborators:

Class: Sync

Description: Manage data transferal from lead and client.

Responsibilities:

1. Allow clients to push to lead. (Settings 3)
2. Allow clients to pull from lead. (Settings 3)
3. Stores pending push requests from clients.

Collaborators:

- Settings

CRC	Keikaku 企画	Date 3/2/2020 9:26 PM	Page 9
-----	------------	--------------------------	-----------

Class: SplunkManager

Description: Interface between Splunk and the system to submit log files and retrieve log entries.

Responsibilities:

1. Sends files to Splunk for log ingestion. (LogFile 1, 2, 3, 5)
2. Retrieve parsed log entries. (LogEntry 2, 3)

Collaborators:

- LogFile
- LogEntry

[END]