1 Hyades Collection Engine (HCE) Requirements

This requirements document is for the Hyades Collection Engine (HCE) sub-project. This is a living document, and will be updated throughout the life of the project.

This document is intended for the following purposes:
- A reference document for designing the system.
- A development and integration resource.
- Identifying and clarifying assumptions, dependencies, requirements and design guidelines.

In addition, it also tries to address the question of compatibility and migration issues of existing agents and client applications resulting from the change in HCE.

Please note that these requirements may not be all implemented immediately in the next release but all functional design should take these requirements into considerations for immediate or future support.

1.1 Revision history

Rev.	Date	Author	Summary of Changes
0.1	September 7, 2004	H.M. Nguyen	First draft

1.2 Functional Requirements

This section lists requirements by categories.

1.2.1 Performance

#	Description	Usage
1	Ability to support unparsed binary data format	Avoid XML parsing overhead for every request.
2.	Small footprint & small CPU utilization during the data collection process.	Be as little intrusive as possible especially during data collection.

#	Description	Usage
	We can define specific metrics for different types of usage or applications.	

1.2.2 Multi-product and multi-user support

#	Description	Usage
1	Ability to run multiple instances of HCEs within the same system	Each product can package its own instance of the HCE. Multiple HCEs can be started within the same system at the same time.
2	Ability to run in shared mode where multiple products are sharing the same single instance of HCE.	This also implies the HCE can accept requests from many different client applications.
3	Ability for an agent to interact with multiple HCEs on the same system	

1.2.3 Scalability

#	Description	Usage
1	Ability to connect and manage a large number of systems and agents.	This implies the concept of controlling and sub-controlling structures where complexity can be managed by breaking down into smaller sets.

#	Description	Usage
2	Ability to connect to other HCEs	A network of HCEs on many different systems

1.2.4 Communication

#	Description	Usage
1	Ability to easily add support of different communication protocols through a pluggable interface.	Initially, we will implement only socket and HTTP. Other communication mechanisms can be included such as message queue or proprietary protocols.
2	Ability to accept multiple concurrent connections of same or different types.	Hyades engine should be able to accept requests from multiple or different connection points: socket, HTTP, HTTPS at the same time.
3	Ability to provide connection timeout	To be able to time out a certain connection for “send” or “receive” operations.

1.2.5 Command and data filtering & format support

#	Description	Usage
1	Ability to manipulate or edit command or data before the agent receives and processes the request.	Be able to edit data prior to processing it (e.g. UUEncode, SOAP processing, etc.)

#	Description	Usage
2	Ability to perform data compression/decompression	
3	Ability to perform data consistency check	
4	Ability to support double-byte characters and localization	

1.2.6 Search, discovery and directory services

#	Description	Usage
1	Ability to determine which agents are available (including all agents that are not currently running)	
2	Ability to discover and retrieve agent's metadata	
3	Ability to search for specific agents	
4	Ability to group a set of related agents	

1.2.7 Security

#	Description	Usage
1	Ability to authenticate at connection establishment	Authentication is performed only once. Subsequent requests on the same connection will not need to be authenticated.
2	Ability to authenticate for each request	Used when the system is loosely coupled (e.g. SOAP message)
3	Ability to authenticate at different security levels: - None - OS user management - Digital certificate	
4	Ability to encrypt data during transmission	SSL support
5	Ability to launch agents and applications under different user-specific security context	

1.2.8 Asynchronous event and error support

#	Description	Usage
1	The agent should be able to communicate with the client asynchronously (e.g. notify when the	

#	Description	Usage
	data collection is finished or an unexpected error occurs)	

1.2.9 Common Services

#	Description	Usage
1	Data/file transfer between the client and the HCE	It must be bi-directional. Either side can initiate the data transfer.
2	Process launcher and the ability to customize the launching process	The customization is needed for agents that perform pre and post processing such as binary instrumentation before launching.
3.	Environmental info	This can be common or platform-specific info

1.2.10 Agent/data collector development

#	Description	Usage
1	Dynamic load of the agent	The ability to add new agents without recompiling or even restarting the HCE.
2	The ability to provide the dynamic interface to query and perform task	

#	Description	Usage
	programmatically	
3	Simplify the process of developing agents/data collector	The ability to share common base code and implementation. The plug-in architecture and the pipe-like development to customize processing based upon the user-specific needs
4	Ability to share same interface between different agent providers	Interoperability between products. Each provider can simply use the same interface but have its own implementation.

1.2.11 New platform support

#	Description	Usage
1	Ability to support 64-bit systems.	The ability to support different sizes of the same data types on different platform.

1.2.12 SOAP and Web Services

#	Description	Usage
1	Ability to support the request/response as a SOAP message	
2	Ability for the agent to publish its own	

#	Description	Usage
	interface in the WSDL format	

It is noted that the design framework should be able to support both "binary" and "XML" formats and let each individual product decide which format to adopt for its own specific requirements.

1.2.13 Language support

#	Description	Usage
1	Ability to support non-java language (e.g. C/C++) for the client and agent	

1.2.14 Hand-held device support

#	Description	Usage
1	Ability to perform data collection on different operating systems for handheld devices such as WindowsCE, PalmOS and Symbian.	

1.2.15 Persistence and fault-tolerance support

#	Description	Usage
1	Ability for the client to disconnect and resume the operation at a later time while the agent is still running	
2	Ability for the HCE to persist its state	The HCE could provide a mechanism for

#	Description	Usage
	and restart.	"lost connection" notification and being able to re-establish the connection between the client and the collector when it is up and running again.

1.2.16 Logging, diagnostics and serviceability support

#	Description	Usage
1	Ability to log for all events and errors	Be able to record important events, usages and who. Be able to re-direct log info to any specified destination (e.g. file, data base or a socket program)
2	Ability to capture all error info into a separate destination	

1.2.17 Agent compatibility

This lists of different options for migration purposes. We need to understand the implication, pick one and plan for it.

#	Description	Usage
1	Existing agents will work without change. The implication is that the HCE will support both types of agents.	Cons: - HCE will be more complex

#	Description	Usage
2.	A wrapper code needs to be developed to support existing agents.	Cons: - Unnecessary overhead
3.	Existing agents will be required to be re-written to adopt the new HCE.	Cons: - Require release coordination and development time.

1.2.18 Client application compatibility

This lists options for the Hyades client side (e.g. Eclipse plug-ins). We need to understand the implication, pick one and plan for it.

#	Description	Usage
1	Existing client will work without change by re-implementing the Hyades client library/packages. Implication: client library needs to detect which version of the HCEs and build the request accordingly.	Cons: - Service pack for the client library
2	Existing client will work without change including current Hyades client library. Implication: no change on the client side but the HCE needs to handle different versions of the client	Cons: - HCE will handle different versions of the client