1 Executive Summary

The Reality Warp Software (RWS) team brings personal experience and lessons-learned from architecting and developing the DCGS-Army's current fusion framework and the Army's first cloud-enabled, mobile, biometric platform. This personal experience gained by Reality Warp Software's founder and President as the Chief Architect for these prior solutions provides a unique perspective and a promising initial direction to Reality Warp Software in providing distributed, scalable frameworks for Big Data analytics. These experiences have lead to the architecting and development of a different kind of technology in the Pirkolator that allows RWS to face the challenges of a complex, multi-modal enterprise such as the one that the Air Force represents with its many and varied sister agencies.

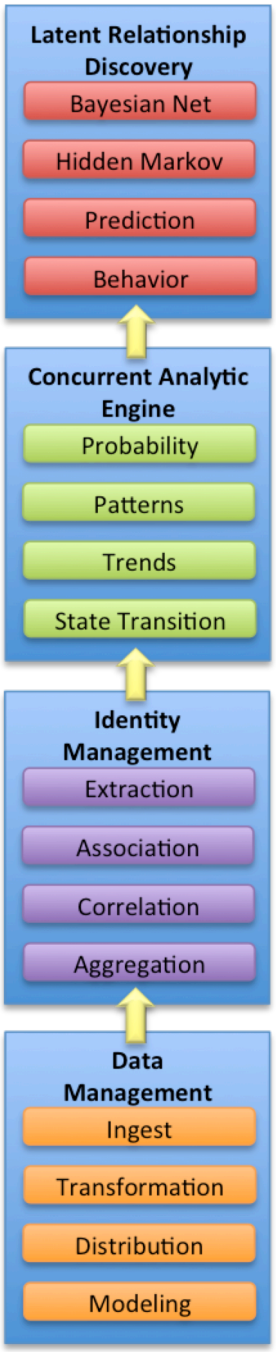


Figure 1: Steps to Discovery

Reality Warp Software has developed several tools that provide us the capability to rapidly developing analytics. We have built the Pirkolator, which provides a distributed, scalable and extensible communication infrastructure that is the foundation for analytics. We have built in support for data ingest that supports many formats, such as JSON, compression techniques, and security protocols, such as PGP. We provide built-in support to ingest from a variety of different data source types from message queues using AMQP to cloud sources like Amazon's Web Services to persistent stores like relational databases and NoSQL stores. The infrastructure provides a transformation library that can automatically transform ingested data into other forms. We provide an abstracted data layer component that is used to provide access, search, query, and changes to specific data source implementations. Using this infrastructure, we built the Determinator to provide a computation engine for parallel processing targeted for use with association, correlation, and aggregation analytics. We built the Trendinator to provide an engine and tools for managing analytics work with data in series, such as over time. We have built a distributed management capability to load target data into these engines to prevent duplicate computations and enforce concurrent access when analytics work with dynamic, changing data. We use these tools to explore new ways to think about data.

The Phase I opportunity under this topic will give Reality Warp Software the chance to explore using our infrastructure, engines, and tool kits for latent relationship discovery. We will explore the use of patterns, state transition models, and the hidden Markov model for extracting relationships based on probability analysis using existing data sources. We are currently developing an engine based on implementing the hidden Markov model to generate probabilities and explore patterns of behavior against market data. We will use this experience for the Phase I effort, to research and develop building the hidden Markov models to provide a computation platform for textual and

Benefits of Reality Warp Software's Architecture
• **Adaptability for mission modifications as threats change**
• **Interoperability to leverage all aspects of enterprise for developing mission strategies**
• **Extensibility to adapt to new sensors, data, and services**
• **Flexibility to build on past mission successes to formulate new missions plans with best options**
• **Ease of use for last minute modifications in dynamic, evolving scenarios**
• **Distributed to connect all available resources**
• **Scalable for optimized analytics and resource utilization**

non-textual data sources. As part of our Phase I delivery we provide the latent relationship pattern and discovery analytics developed with this platform. Given the opportunity to continue under the Phase I Continuation, we will use this time to solidify the target data sources, being extension of our technique to broader, distributed Bayesian belief net, and deploy to a cloud cluster as we start the transition to a Phase II deliverable. We anticipate our Phase II deliverable will consist of providing an extensive, distributed Bayesian belief net combined with our pattern engine and hidden Markov probability model. We anticipate deploying our infrastructure to a cloud cluster using customer specified data sources.

Reality Warp Software's approach to discovery and analytics is a 3-step solution. Our first step is to data modeling and acquisition. We integrate the disparate systems, services, and data sources using our distributed, scalable architecture. Our next step is to provide Entity, Event, and Relationship extraction, meta-data association from data ingest, and the Determinator engine to provide highly resolved, temporally and geospatially defined entities and data segments. The final step in our approach is to use the highly resolved data to feed our Trendinator engine and other collaborative inference analytics to discover the latent relationships and threats that are buried like the proverbial "needle in a Big Data haystack". For this opportunity, RWS will leverage existing tools to allow us to focus on the topic of latent relationship discovery and new techniques and analytics to extract such information.

For Phase I, our main focus will be the exploration of the hidden Markov model to generate probabilities that can be used for latent relationship extraction. We will leverage our Trendinator engine to apply entity, pattern, and probability computations run over a series of elements, either in time or another sequence type. While not the focus of our Phase I proposal, we believe that establishing highly resolved data is important to produce accurate probabilities. We will provide modeling around the concept of Identity Management that is used to assert the accuracy or establish the truth of a data set. We will use this concept to model the attributes of entities, events, and relationships that compose an Identity. Derived metadata on highly resolved entities provides analytic solution developers with information about the type of system and data formats used when acquiring entities. Data formats, such as JSON or XML, and system information, such as operating system, are associated to each extracted entity. Pedigree provides for resolution of where data comes from, how it has changed over time, and the sources that have affected the change. Being able to traverse pedigree allows for analytics to build accuracy, confidence, and assurance trees for prioritizing similar data. Contextual metadata allows for categorization and classification of data. The benefit of contextual metadata is in filtering the data before processing. Knowing that an entity does not contain geospatial data prevents the data from being distributed to a geospatial-based analytic. Using all this information in a common, efficient platform using economies of scale, we will provide in Phase I a near real-time synthesis of data using our Determinator engine for associated, correlated, and aggregated resolution of

entities and our Trendinator engine for establish the computation, pattern, and probability analytics.

2 Technical Approach

2.1 Latent Relationships

Reality Warp Software proposes to use a 3-step process to research and develop analytics for latent relationship discovery using state-of-the-art techniques. We will develop these analytics against the problem of identity management that has a wide range of applications. First, deconstruction of multiple data sources into elements will be explored to produce an analytic model with extensible attributes targeted for efficient processing. We will use the deconstructed elements to train probability models from existing data and build patterns based on these probabilities and other known pattern types. Finally, we will apply the trained probabilities and generated patterns using two different probability models. We will use a hidden Markov model to deduce probability of occurrence for known states and transitions on new observations. Given the opportunity to continue under the Phase I continuation, we will use a probabilistic directed acyclic graph model (Bayesian network) to build and maintain identity elements against the potentially related relationships.

The hidden Markov and Bayesian network models are widely understood models, but are not the only models that RWS will be able to explore. The key to our research and development lies in the way that we will approach the discovery process. We will leverage an existing open-source, distributed, scalable computation platform to perform our analysis. On top of this we will apply extraction, decomposition, and analytic modules in a way that will allow for efficient use of distributed resources and provide for computations in parallel. By leveraging this infrastructure we will be able to utilize larger amounts of data using new techniques of in-memory data grids for real-time discovery.

For this proposal we will describe three areas of research and development that will encompass the aspects of this topic – data positioning, analytic models, and the computation platform. Data positioning involves describing the data efficiently to be used by the analytic models. This includes ingest from data sources and entity extraction. We will focus on two analytic models – the hidden Markov and the Bayesian Network model. The computation platform answers the "how" we're going to use parallel, scalable, and distributed analytics applied to the Markov and Bayesian models.

We will start by describing the concept of Identity Management that forms the "what" aspect of the relationships that we will be targeting in the results of our research and development. We will explore several computational models and analytics we will be use to implement Identity Management. These strategies will be explored as we build the modules within our Markov and Bayesian models. This will be the main emphasis of our research and development for this opportunity. Lastly, we will describe the supporting platforms, engines, models, and other tools necessary to build an application of latent relationship discovery in real-time that RWS can provide. These consist largely of existing, open source tools that will be provided at no cost or development time under this effort.

2.2 Identity Management

Identity Management describes the processes and techniques used to positively or negatively establish an entity's "true" identity. An entity is anything that has describable attributes. Typical entities involved in Identity Management are Person, Event, and Place.

Attributes are used to describe an entity. For a Person, there may be a name, social security, or hair color attribute that describes the given entity. Entity relationships are used to describe attributes that exist between two or more entities. For example, "marriage" is a relationship that exists between two Person entities; "attended" is a relationship that exists between Person and Event entities. An Identity is applied to an entity and describes attributes and relationships that are needed to assert the accuracy of identification.

Typically, Identities are applied to Person entity types and include ascribed and forensic attributes. Ascribed attributes describe characteristics of an entity that have been given to that entity, such as a name or social security number. Forensic attributes describe inherent, measurable characteristics of an entity, such as hair color, fingerprints, or DNA. Each of these types of attributes can be used to assert the Identity of the entity. Implied relationships can also be used to assert the Identity of an entity by describing a relationship condition between the Identities in question and another known entity. An implied Identity could be a proximity relationship, either temporal or geo-spatial, that exists between two entities. We will use these concepts to develop a model that we can use for latent relationship extraction in Phase I.

2.3 Analytics for Latent Relationship Discovery

RWS has identified and will explore several computational models and analytics in support of this effort. Computational models are distinctly different ways to perform discovery – the "how do you do it" art of the possible. RWS will develop tools and methodologies for applications of each of these models. Computational analytics provide the means for discovery – the "what do you do" art of the possible. Additionally, RWS will explore utilizing existing contextual, syntactical, and meta-data processes to support enhancements and efficiencies to the latent relationship discovery process.

2.3.1 Computational Models

We have identified several computation models that we will explore for latent relationship discovery. We propose to start with very simple, single entity computation prototype to provide the proof of concept for this effort. Additionally, we propose to leverage our existing distributed, parallel and scalable computation framework, analytic tools, and data extraction framework to provide this prototype in a rapid fashion. With the computation infrastructure in place, we can focus on the more difficult problems of predictive and pattern computations that require scaled and parallel-processing resources.

2.3.1.1 States and Transitions

States are defined given static, known definitions. Dynamic states are special state definitions that are derived from new sets of data. Known states are those that can be described before building a computation algorithm. An example of a known state would be the closing price of a company's stock or as simple as the "off" and "on" states of a light switch. Dynamic states are those that become apparent given new data. For instance, in exploring textual data, the first step is to dissect text into words and lemmas. A lemma is a phrase or word that has been tagged with a specific connotation, such as "I want to bomb a coffee shop" with the tagging being "terrorist event". The new combinations of words and lemmas define new, previous unknown states. These dynamic states can then be added to the analytic process during discovery.

States can be further utilized by observing the transitions that occur as states change. Transitions can be used to develop patterns and behaviors that enhance the discovery process. In

the stock market, a simple transition would be the change in a company's stock price and can be quantified by the amount of change in the price. Likewise, the change of a Person entity's place of residence would also establish a transition in the state defined for residency. States and transitions define a simple organizational architecture that forms the basis for applying analytics for latent relationship discovery. For Phase I, RWS will define states and transitions related to Identity Management and apply these to the computation methods and analytics defined in this proposal.

2.3.1.2 Single-Attribute Computation

Single-attribute computations involve a series of discrete, similar attributes applied against an analytic to compute a result. Discrete attributes typically contain a single value, such as a name, signal frequency, or other discretely measured phenomena. Computations are performed on a single subject attribute against a single target attribute to compute a result. An example of a single-attribute computation would be the application of the Levenshtein distance algorithm to compare subject A's name with target B's name. The result would be a relationship that describes this result, such as a "same as" or "different from" relationship.

2.3.1.3 Single-Entity Computation

Like single-attribute computations, single-entity computations involve a series of discrete, similar entities applied against an analytic to compute a result. Discrete entities typically contain a single entity type, such as a Person or Event. Discrete entities could also be attributed data that is common across entity types, such as geospatial or temporal attributes, to determine results across different entity types. Computations are performed on a single subject entity against a single target entity to compute some result. An example of a single entity computation might be the location and time attributes of a Person entity computed against the location and time attributes of an Event. The result would be a relationship that describes this result, such as a "same location", "same time", or "participated in" relationship.

2.3.1.4 Predictive Computation

A series of discrete, similar entities are applied in order against an analytic to compute a result. This result is used to identify the static state that exists between each sequence of entities in the series. Each result in the series is tallied to produce a probability of occurrence for each identified static state. Prediction is used against the probability to determine the likelihood of the next occurrence or state of an entity. This is the basis for applying the hidden Markov model to predict behavior without having to understand the behavior being modeled, just having awareness of historical behavior to calculate likely, new behaviors. For example, given state A, if 90% of the time it is observed that there is a transition to state B with the other 10% of the time being a transition to state C, one can then predict with a 90% chance of accuracy that given state A, the next transition will be to state B.

2.3.1.5 Pattern Computation

Pattern computations extract relationships against patterns of entities. Static patterns extract relationships based on a match against a discrete subject and target entity or event. State patterns extract relationships based on a match against a series state of a subject and target entity or event. Linear state patterns match a series of subject states against a target series of states. Transitional state patterns extend linear pattern recognition to compute the deltas between values of a linear subject state series and a target state. Predictive state patterns match on the computed

probability that given a series of subject states that a target state or target pattern will next occur. Predictive state patterns use the same concepts of Predictive Computation described above.

The first step in developing a Pattern Computation is defining the states and transitions. The next step is developing the permutations of interest. Complete permutations allow for duplicate states, while unique permutations only allow one state per pattern set. Constrained permutations allow for definition of specific combinations of states with a pattern set. In addition to the permutation being applied, the pattern set size determines how many states are sequenced in the pattern set. An important reminder in developing pattern-based computations is the number of permutations that can result. This can be as large as n^r with n being the number of states and r being the pattern size. Given five states and a pattern size of ten, there are 9765625 permutations that can be computed.

Matching patterns against potentially large numbers of permutations requires a capability to maximize utilization of all available resources. Distributed and parallel computation techniques are two ways of attaining this utilization. Distributed and parallel computation relies on an infrastructure to manage computation requests, establish methods for concurrent access of dynamic data, and promote results to other analytics, persistence layers or visualizations.

For Phase I, Reality Warp Software will leverage our existing infrastructure and computational engines to break computations into modules that can be run in parallel and distributed. Computational modules contain the states, transitions, patterns, and any other required data needed to perform a single computation of a subject set against a target set. We will apply an existing tool to manage loading the computational modules across many computers. This tool prevents duplication of processing and also provides concurrency management for dynamic data sets that require real-time changes to computation modules. We use an in-memory, distributed data grid to provide the registration of available computational modules and notification of data set changes that need to be propagated through the cluster of computational modules. An example that we will explore for Phase I using textual data will involve matching lemma patterns from Twitter data. Given 1000 different combinations of lemmas of interest and 10 systems running analytics, we can break the 1000 lemmas across each of the 10 systems, so that a single system can match against 100 of the lemmas. When a new set of text becomes available, a request with the text is distributed to each of the 10 systems. Each system computes the results of the new, subject data against the 100 defined, target lemmas. The results for each of the 10 systems can then be distributed to any other analytic interested in the result or to an aggregator that collects all the results from a single request.

Matching data sets using pattern-based computations may not *qualify* the pattern of the data, but it does quantify that the pattern exists. Applying event association to the pattern development provides the first step to *qualifying* the result of a pattern match. Distributed, parallel processing of pattern matching gets us close to being able to discover latent relationships in real time given enough resources to perform the computations, but only allows us to establish and quantify the number of times patterns occur over data sets and streams. Where the real power of this architecture comes into play is in associated events or transitions to patterns. If we can define the transitional state of the next occurrence from a pattern, we can establish a prediction that given a pattern occurrence, the next step will likely be some next state. For example, continuing with our lemma scenario, given that we have a new Tweet that matches the "I want to bomb a coffee shop", we may be able to predict that the next state might be "I need to make a bomb". While a simplified example, this insight is the start and basis for discovery of latent relationships through real-time, computational analytics.

2.3.2 Computational Analytics

Computational Analytics provide the foundation of this research and development effort. RWS proposes to explore and develop analytics that cover three main analytic result types. From this exploration, we will develop tools and techniques that can be applied to most data sets and algorithms for latent relationship discovery forms. This will allow us to rapidly develop capability for any latent relationship types need to support current mission sets.

2.3.2.1 Probability-based Analytics

Statistical probability inference analytics use the results of computations to assign probabilities. Computation of probabilities, while a potentially resource intensive operation, is a simple mathematical function. The real power of probabilities is through the inference that can be established given the resulting calculation of a probability. An understanding of the mathematical function is necessary to ascribe the inference type that can be implied to a data set. There are many inference types and for this opportunity, we will focus on accuracy, similarity, and occurrence types. Statistical accuracy can be derived through probabilities computed against similar data from many data sources. For instance, given three data sources A, B, and C each with an entity X that contains several attributes, a computation is performed to match the attribute values across the three data sources. Suppose that both A and B compute that all the attributes are equivalent, but C computes that not all the attributes match. Given that each data source is equally trusted, you can infer that entity X on data source C may not be 100% accurate. Similarity probabilities produce measurements of the likeness between sets of data in a fashion similar to accuracy probabilities. Similarity computations rely on known configurations and other analytics to provide a determination of what is similar and what is not. Entity de-duplication across data sources is an example of the application of a similarity inference. Occurrence inferences are used to establish order and then predictions for the next iteration or outcome given some new data. Occurrence probabilities are applied to discover the likelihood, based on existing data that a behavior will occur. For instance, if you apply 3 states to the closing value of a company's closing market price – low, no change, and high – you can determine, based on previous performance, the likelihood that the next price will be lower, no change, or higher. This is also an example of the application of the hidden Markov model.

2.3.2.2 Behavior-based Analytics

Behavior analytics associate behavior conditions to data sets. Sentiment analysis is a very common form of behavior analysis that associates negative or positive connotation to a corpus of data based on language or some other perceived conditional value (like stock market price falling or increased number of terror events). Behavior analytics are not the focus of this opportunity, but we will use sentiment analysis combined with probability to extract behavior-based predictions. For instance, we can establish the sentiment of Twitter data over time to establish a statistical measure of the sentiment of all tweets over some interval. We can then explore sentiment patterns to find relationships that might indicate some change. For instance, if the daily sentiment over the last week was constant at around 60% positive sentiment and tomorrow the overall sentiment drops to 40%, then some occurrence or relationship could be established.

2.4 Risk Analysis

While a complex topic, RWS believes that the risks in performing this analysis are minimal. The biggest risk in this analysis is the availability of a quality corpus of data. To

mitigate this risk, RWS can provide generation of test data using publically available sources. Additionally, RWS can provide the capability to consistently re-play data to simulate real time, streaming data sources. Another risk that RWS has identified is in the applicability of the results. It may be the case when we start exploration of this topic that the results do not lend to latent relationship discovery. To mitigate this risk we will provide weekly progress results and establish an open forum of communication to ensure that our algorithms and approach are in line with the expectations of the customer. Additionally, if available, we work with other experts in collaboration to provide the most effective results.

2.5 Existing Infrastructure and Tools

To support the effort of researching and developing advanced, state-of-the-art latent relationship analytics, there are several supporting fundamentals that make this process much easier and faster. In the following sections, we will describe each supporting capability or tool as data moves from the originating source to distributed analytics and then being represented as views to analysts.

2.5.1 Infrastructure

Reality Warp Software has developed an open source, distributed analytic infrastructure, illustrated in Figure 2 included at the end of this proposal, that provides the system platform and key tools enabling distributed data processing. This infrastructure enables analytic developers by providing access to real-time and persistent data streams, efficient use of dynamic resources, and dissemination of results to an ever changing and evolving community of consumers. The infrastructure is architected around the publication/subscription pattern to establish a communication and data distribution platform that is tailored to processing real-time data streams, persistent SQL and NoSQL data stores. We recognize the ever-changing data landscape and provide analytic writers with a stable way to access data without worrying about the changes to the underlying data sources. The infrastructure addresses this by providing an abstracted data access layer that analytic developers use to access required data. The infrastructure inherently supports several architectural patterns for analytic execution. The distributed consumer pattern is used to provide the same data to analytics executing in parallel. The competing consumer pattern is used to execute many of the same types of analytics against streams of data in parallel. The event-driven consumer pattern is used to execute analytics when events and other stimuli occur. For this opportunity, we will leverage the benefits of this existing infrastructure for easier and timelier research and analysis of the efficiency of data processing and analytic results. Additionally, by leveraging this infrastructure, RWS will be able to more rapidly develop, prototype, and test analytics.

In building this infrastructure, we leveraged the experience gained from architecting and building the fusion infrastructure currently integrated with the DCGS-Army program. This insight and experience paved the way for the creation of an infrastructure that is tailored to running analytics under a variety of conditions. By leveraging open source and open standards, we have built-in compatibility with this and other standards-based infrastructures. We built a publication and subscription infrastructure that is both scalable and distributed. We utilize an open source application, Hazelcast, to provide one mechanism for distributed pub/sub, but the infrastructure is not limited to just Hazelcast. The Pirkolator is able to incorporate additional distributed pub/sub applications, such as Redis or Terracota, to provide the concept of a “Hub” for any pub/sub need. We incorporated the open source data transfer application, Apache Camel, to provide access to the most common network protocols, data formats, and translations. The

benefit of this approach is more consistent, timely access to data and greater assurance of data integrity. For this effort, we will leverage the flexibility of this environment to incorporate existing open source analytics, tools, or other applications that will enhance our research and development of new and more efficient processing of latent relationship techniques.

2.5.2 Data Ingest and Extraction

Data streams and data ingest is provide through another open source application from Apache called Camel. This product provides us with an extensive range of data protocols and formats, as well as transformation services to position the data in an optimized way for analytics. We utilize Spring Data, an open-source platform, to provide an extensible, flexible, and abstracted data management technology. This technology allows us to access to out-of-the-box implementations for several data modeling standards, such as Relational Database Management Systems (RDBMS), Big Data, and Key-Value Stores. Custom repositories allow us to develop for any data source need. Using an abstract data access layer coupled with the open source Spring Data technology, allows analytics to focus on their job without worrying about changing data source environments. This means no code re-writes should a data source or even data format change.

2.5.3 Data Modeling

Data modeling provides for describing data in a digital form. From the personal experience of the Reality Warp Software team trying to provide a common model to describe everything is not feasible. We believe that a better, more efficient and extensible solution is to develop models that a tailored for how the data will be used. Analytics have different modeling needs than persistent storage and a signals processing model has different needs than textual processing models. A common data model is used in cases where different systems need to work together to share data in which case the model is developed around the common model attributes.

Using multiple data models, tailored to specific needs also increases the extensibility as new data types and models becomes available. This flexibility manifests through addition of new fields to an existing model and through addition of new data models working in concert. Some data models are not extensible in which case new data models must be developed to work in concert.

For Reality Warp Software, data modeling's most important benefit is in positioning data for optimal utilization. By using tailored data models and architecture of extensibility, we are able to position data for processing even as the data model changes. To implement this strategy, though, data modeling is not the only function needed. The addition of transformation techniques combined with our thinking about data modeling allows us a novel way to position data that can be leveraged across systems and under a variety of analytic processing.

2.5.4 Data Transformation

Transformation services ensure that data is mapped and translated into the optimized format used by each analytic. Filtering methodologies built into the publication/subscription construct provides even further data refinement for analytics. Geospatial, temporal, and source-based filters provide analytics with data granularity, reducing the size of the data haystack. Custom filtering is provided to allow each analytic with data filtering tailored specifically to its needs. Our transformation services utilize the open source technology, Apache Camel, to provide network connectivity, data routing and common data format transformation support, such as for JSON and XML. We provide additional hooks into this data transfer framework to provide

dynamic mapping and transformation of incoming and outgoing data to support any object model needed. The benefits of this approach are extensibility, reusability, and flexibility. As the data sources within the community change, extensibility within the infrastructure provides new transformation services that are automatically integrated without affecting analytics or other data-aware processes. We are able to reuse existing data mappings and transformation services where needed and the modular nature of new transformation services allow portability to other systems outside of the systems integration effort. The flexibility of the transformation services within the infrastructure allows for analytics to get the data in the format needed without having to understand the underlying ingest or originating format of the data.

2.5.5 Data Management

The data management component provides the necessary architecture for plugging in a variety of data stores by abstracting away the details of data management. We utilize open source technologies, such as Hibernate and iBatis, to provide data modeling, persistence, and query capabilities to structured RDBMS and Object-Relational Model (ORM) schemas. Unstructured data or large amounts of data are stored in a NoSQL data stores, such as Mongo DB, Accumulo and HDFS. It is important to allow for the integration of Big Data solutions and to provide the ability to run analytics over massive amounts of data in near real time. Multiple data source solutions are optimized for storing different types of data. With the ability to integrate these disparate data sources, analytics are optimized when performing complex operations specific to a data source, such as entity correlation and complex event detection.

Our infrastructure provides integrated metrics, pedigree, and data integrity. Metrics lend support to the systems administration command to maintain a healthy platform. Pedigree is integrated with the infrastructure to allow tracking of the processes that change or affect data. This builds upon the confidence, accuracy, and other data integrity definitions that help establish a reliable data tree of origination for each piece of original data, extracted entity, or transformed data. Having this tightly coupled relationship between pedigree and analytics promotes a strong chain of authentication when data verification is needed.

Our strategy to data within the infrastructure provides several benefits. First, it allows for dynamic data sources. As data sources come and go online, analytics don't have to change. Transformation services ensure that data is in the format needed for an analytic. Pedigree built into the fusion infrastructure further ensures data integrity, insight, confidence, accuracy, and traceability. With cloud technology and cheap compute cycles, transformation services allow data to take many forms and support many models. If you need to map data from one data model to another, transformation services provide an easy mechanism to achieve this goal. Every analytic can have its own optimized data model while at the same time support a common data model that is used across enterprises.

2.5.6 Distributed Analytics

On top of these open source applications, Reality Warp Software provides easy to use architectural patterns that allow other processes, such as trending analytics or map/reduce functions, to get the data, events, or manual stimuli needed for a solution. We employ new techniques and technologies to achieve the near real-time analytics over Big Data sources that are the first step in developing a real-time, latent relationship discovery capability. We employ an in-memory data grid that consists of Java Virtual Machines (JVMs) running on any number of servers in the data center. An in-memory data grid is a distributed memory based data store that allows the data to be distributed across many servers. This is backed by disk storage with a write

behind scheme for permanent storage. The benefit of this technique is to allow fast access to large amounts of data, even under systems with load constraints, while still providing the needed streams for real-time analytics. The in-memory data grid can be implemented with off-heap storage to utilize larger amounts of Random Access Memory than is allocated for the JVM, and to avoid the penalty of the garbage collection when items are deleted. This new technique to in-memory, distributed data affords the capability to explore computational analytics against real-time data streams for producing real-time insights.

New technologies allow us to provide parallel, concurrent processing, in-memory and distributed. This can be much faster than traditional disk-based Hadoop/HDFS infrastructures. Additionally, concurrent technologies will promote new thinking. From past experience, a new way of thinking about an analytic was identified as an economy of scale for a CERDEC analytic. The Communication Effects Simulator (CES) was identified as an important analytic, but was slow due to being a Windows command-line tool that must be manually run. To achieve benefits, it was possible to run the analytic in parallel and then combine the results. This allowed for a 40% improvement in working with the analytic.

For legacy analytics that are Hadoop-based, we can leverage the results in the same way we distribute data by taking the output of a map/reduce job and sending it to where it needs to go. We use a persistent cache within our in-memory infrastructure to provide fail-safe capability in the case of system failures. Additionally, we are able to leverage large disk-based systems, just as if they were in-memory sources when the amount of data is too large to keep in memory. This type of in-memory, distributed processing is at the heart of where we think future processing platforms are heading. We are integrated with another open-source application called Spring XD. This platform provides a "unified, distributed, and extensible system for data ingestion, real time analytics, batch processing, and data export". It provides out-of-the-box access to Hadoop operations, such as Map/Reduce and HDFS. This is just another example of a tool that is easily integrated into the infrastructure that keeps us at the cutting edge of analytic development.

Trending analytics provide one of the two bases for exploration in this proposal. The infrastructure, data modeling, and other analytics support this type of analytic. Trending analytics allows for collecting data to allow inference discovery across groups of data. Aggregation and correlation are fundamental properties of trending analytics. We propose to use our existing Determinator engine, made up of a combination of aggregation and correlation analytics, to develop trends against textual and non-textual data in the Identity Management problem. These trends will contribute to discovery of latent relationships through development of predictions and probabilities that can be applied within our implementation of the hidden Markov model being explored in Phase I of this effort. Additionally, the resulting trends will contribute to the establishment of the distributed Bayesian belief network used to identify confidence that a trend exists on seemingly random, unrelated sets of data in future Phases of this effort.

2.5.7 Analytic Engines

Reality Warp Software's Determinator engine is a capability we developed to allow algorithms for Entity, Event and Relationship (EER) extractions to combine with Association, Correlation, and Aggregation derivations to provide more resolved entities. Our engine utilizes the distributed and scalable nature of the Pirkolator's infrastructure to provide a real-time, dynamic scoring engine. Algorithms are developed and added to the engine using the same publication and subscription filtering in the Pirkolator to provide the most granular data available.

An extensible scoring construct is used to provide the results that can then be used for automatic threshold alerting or other post-calculation analytics. Additionally, results are automatically distributed to any interested analytic that is attached to the infrastructure. Lastly, results can be automatically persisted to any attached, disk-based data store to provide further analysis using functions such as Map/Reduce. The Determinator engine provides a fast, extensible, and scalable resource that gives developers a fast and easy start to providing EER extractions, associations, correlations, and aggregations.

Reality Warp Software's Trendinator engine is another capability we developed in conjunction with our infrastructure and Determinator engine to provide the grouping and threshold triggering useful during trend identification. Behaviorally, the Trendinator engine functions much the same as the Determinator engine with algorithms being added to the engine to provide scoring against any input data provided through the Pirkolator. The scoring results are run against threshold constraints and post-calculation analytics to provide automatic alerting. The Trendinator engine provides an additional function to the analytic developer through a grouping construct that can be used to build and establish related trend data. The grouping construct provides temporal and geo-spatial groups coupled with other custom, trend-specific constraints, such as area of interest or time range limits, to automatically keep only the data elements that fit the trend grouping. The grouping constructs utilize the in-memory, distributed data grid of the Pirkolator for fast and efficient storage and retrieval of the data elements. The Trendinator engine provides developers with an immediate solution to quickly providing trend results against real-time data streams utilizing the available compute resources.

2.5.8 Analysis Visualization

Reality Warp Software leverages the Model-View-Controller architecture pattern to provide visualizations to the analyst. With our Transformation Services we are able to deliver representations of data in a format that can be easily viewed in any visualization platform, such as in browsers or standalone GUI applications, such as NASA World Wind. As data is produced within the Pirkolator, the Transformation library provides automatic conversion of the data into any view that may be in use. The Transformation library can also be used to manually transform produced data into view representations as needed. This strategy allows an extensible mechanism to provide data to multiple, different visualization platforms where required. As new ways to think about and visualize data are developed, transform modules can be added to support the new perspective.

3 Phase I Work Plan

3.1 Scope

For Phase I we will limit our scope to the development of latent relationship discovery for Identity Management. We will use existing corpus of data from the USAF if available. Otherwise, we will use Twitter feeds as a textual data source and publically available stock market data as our non-textual data. From these two data sources we will develop patterns, sentiment behavior, and probabilities via application of the hidden Markov model. From the established patterns, behaviors, and probabilities we will research, explore, and define the types of latent relationships that become apparent. Lastly, we will iterate over the established, trained analytics to extract the apparent relationships against a different set of similar data to examine the accuracy and applicability of the extracted relationships.

Given the opportunity to pursue the Phase I continuation, we will implement a distributed Bayesian Net to introduce a larger matrix of probabilities. We will also solidify the data sources that are required and explore adding additional data sources to increase the breadth of data being processed. Lastly, we will explore implementation of other data models in addition to the developed Identity Management to increase the scope of relationships in discovery.

3.2 Task Outline

3.2.1 Phase I Tasks

3.2.1.1 Task #1 – Deploy Infrastructure

This is a relatively short task and involves setting up the project and tools to start research.
Estimate: 40 hours

3.2.1.2 Task #2 – Develop Data Sources

This task develops the ingest code and extraction for Twitter streams and numeric data sets. This task involves developing a mechanism to archive data for play back that is needed for accurate, repeated testing and analysis.
Estimate: 40 hours

3.2.1.3 Task #3 – Develop Data Model

This task will run in parallel with Task #2 to define the models needed for exploration. For textual data we will deploy to Elasticsearch and for non-textual data we will use MongoDB.
Estimate: 40 hours

3.2.1.4 Task #4 – Pattern Development

This task represents defining the states and transitions for computations that will be applied within the hidden Markov model.
Estimate: 40 hours

3.2.1.5 Task #5 – Implement Markov model

This task defines the scope of effort for developing the Markov model and implementing the computational models across distributed resources for parallel processing.
Estimate: 320 hours

3.2.1.6 Task #6 – Result Analysis and Refactoring

Based on the results from Task #5, this task defines the research and tweaking that will become apparent from examining the results of our application of the hidden Markov model.
Estimate: 480 hours

3.2.2 Phase I Continuation Tasks

3.2.2.1 Task #7 – Develop Bayesian Net

This task implements a Bayesian Net using the existing platform and tools.
Estimate: 120 hours

3.2.2.2 Task #8 – Solidify Data Sources

This task further develops the existing data sources.
Estimate: 80 hours

3.2.2.3 Task #9 – Develop New Data Sources

This task develops new data sources that will increase the breadth of discovery.
Estimate: 8 hours

3.2.2.4 Task #10 – Develop and Apply New Data Models

This task covers the development of additional data models determined through coordination with the customer. Additionally this task covers the application of the new data models with the existing analytics.
Estimate: 120 hours

3.3 Milestone Schedule

3.3.1 Phase I Milestones

3.3.1.1 Milestone #1 – Platform and Infrastructure

Schedule – end of 2nd month of Phase I
Deliverable – technical review to include progress presentation and demonstration of platform with data source processing and analytic results

3.3.1.2 Milestone #2 – Relationship Discovery

Schedule – end of 4th month of Phase I
Deliverable – technical review to include progress presentation and demonstration of probability and behavior analytics with relationship extraction

3.3.1.3 Milestone #3 – Prototype

Schedule - end of Phase I
Deliverable - technical review to include final report with SF 298 and demonstration of prototype with Markov model and latent relationship discovery

3.3.2 Phase I Continuation Milestones

3.3.2.1 Milestone #4 – Bayesian Net Implementation

Schedule – midway into the continuation
Deliverable – Bayesian Net implemented on existing analytic platform

3.3.2.2 Milestone #5 – Prototype

Schedule – end of continuation period
Deliverable – Update of platform with Bayesian Net, data sources, and data models

3.4 Deliverables

3.4.1 Phase I Deliverables

Deliverable	Schedule	Description
Kick-off Meeting	2nd week	Initial meeting at Wright Air Force Base
Progress Reports	Weekly	Technical progress updates for customer
Technical Review	Every 2 months	Technical presentation review via VTC for each Milestone
Prototype	End of Phase I	Delivery of prototype application supporting Phase I scope

Final Report	End of Phase I	Final report on Phase I scope to include SF 298

3.4.2 Phase I Continuation Deliverables

Deliverable	Schedule	Description
Continuation Kick-off Meeting	1st week of Continuation	Continuation meeting at Wright Air Force Base
Progress Reports	Weekly	Technical progress updates for customer
Technical Review	Every 2 months	Technical presentation review via VTC for each Milestone
Prototype	End of Phase I Continuation	Delivery of updated prototype application supporting Phase I Continuation scope

4 Past Performance

Our experience from acting as the Chief Architect deploying the fusion framework currently used by the Distributed Common Ground System – Army (DCGS-A) program provides us with a unique insight into the requirements of developing fusion and correlation capabilities. The DCGS-Army fusion framework was developed in 2008 to address the need of data de-duplication within the Army's centralized data model called the Tactical Entity Database (TED). From this experience was gained an understanding of the need for efficient resource management, dynamic data connectivity, and distributed information utilization. It was realized that many of the techniques and methodologies employed in the de-duplication effort could be applied across a variety of analytic processes, such as association, correlation, aggregation, and temporal and geo-spatial trending. This also lead to an understanding that providing an integrated infrastructure with core support for data management and analytic processing engines provides a robust and easily deployed enterprise application solution for addressing specific data needs. This was proven at the Army's Empire Challenge in 2011 where the Army's first, cloud-enabled, mobile biometric fusion platform was demonstrated.

Our experience on the Army's Windshear II program in developing and demonstrating the first fully functional, cloud-based, mobile biometric solution at Empire Challenge 2011 further illustrates our awareness of the challenging problems of today's networked world. Windshear II was demonstrated successfully to the Undersecretary of Defense after only 4 months of architecture and development. The platform consisted of a cloud-enabled server in a vehicle that housed a data management capability, biometric analytics for facial recognition, voice recognition, retinal patterns, and personal identification. These elements were all collected using a Motorola's Atrix Android phones. As the data elements were collected from the phones, associations against an HDFS-based graph data store providing processing to alert back to the mobile devices when entities of interest were discovered. With a short timeline for demonstration and a relatively new application platform in Android, it was learned that open source solutions for data connectivity to mobile devices coupled with utilization of the then emerging HTML5 standard allowed a much faster path to integration of the mobile devices to the cloud-enable infrastructure. One of the stunning successes of this demonstration was the delivery of alerts not only to the mobile devices, but also through secure protocols to alert the higher echelons of the Army. This security driven effort provided the acknowledgement that an integrated, distributed, enterprise platform enables secure management of data.

These personal experiences brought to Reality Warp Software provide the key ingredients and foundation for its initial offerings and solution architectures. The Pirkolator,

Determination Engine, and Trending Engine each provide a new, unique, and differentiated solution that provides the basis for RWS's approach to offering Big Data solutions. In providing these solutions as open source to the community, RWS's strategy is to provide industry specific analytic solutions built upon community driven and maintained tools. This allows for a focus on the real problems facing industry and not around the tools required just to get to the start of answering problems.

5 Key Personnel

Jeff Pirkey

- Principal Investigator

EDUCATION

- M.S., Engineering, University of Texas, 2008.
- B.A., Economics, University of Texas, 2004.

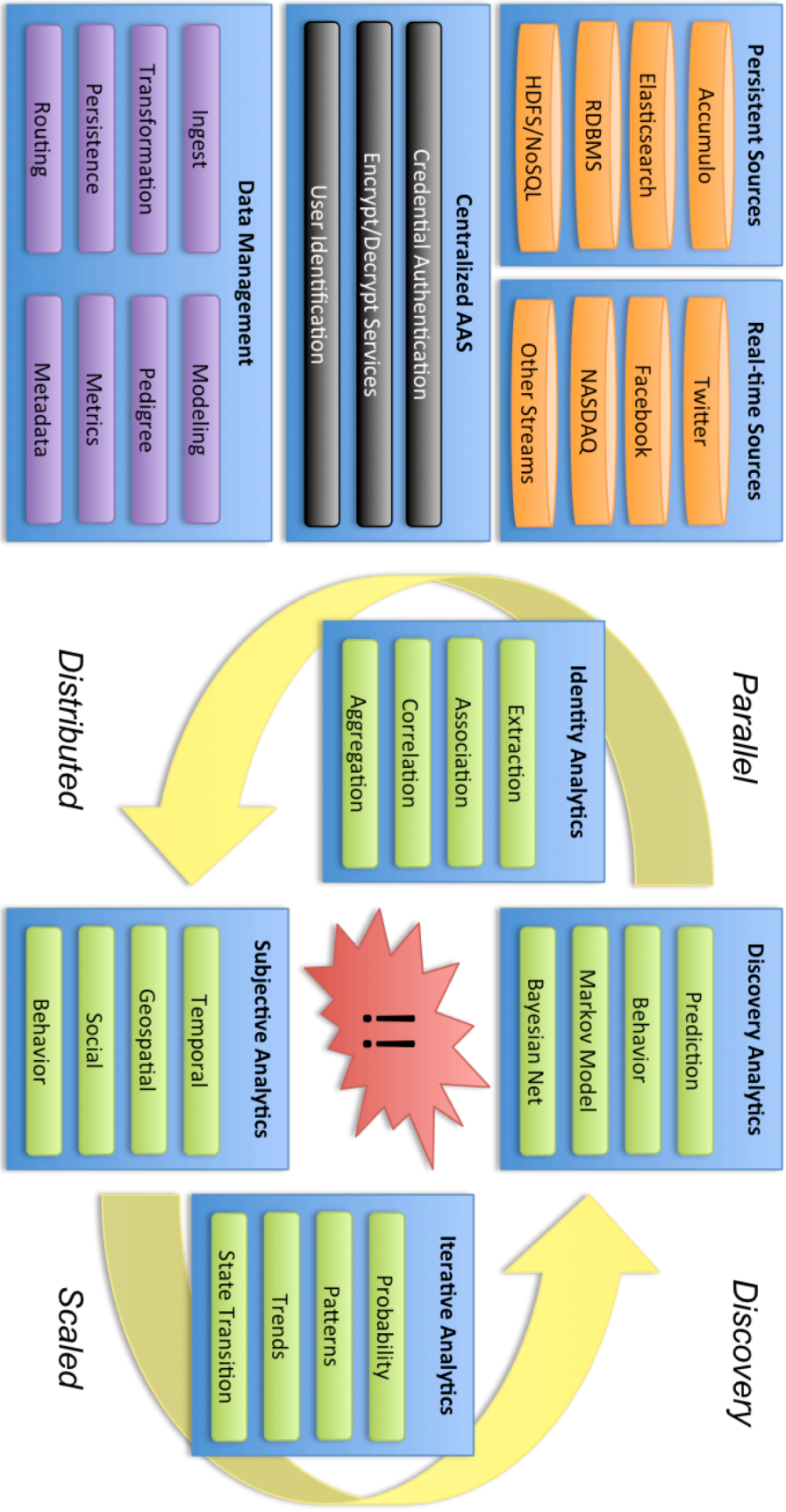
CURRENT POSITION AND RESEARCH

Jeff Pirkey currently acts as Chief Architect for Reality Warp Software and will be the Principal Investigator for this opportunity.

RELEVANT EXPERIENCE

Jeff Pirkey has over 20 years of experience in software engineering with Java and .NET-based technologies. He is an architect, manager, and developer with expertise in stand-alone, distributed, web-based, and visualization-based applications. He architected the white paper recognized at the 2012 Army Electronic Warfare industry day as the necessary solution for bringing the EWO into the fight. He lead the team that developed one of the first successful Army cloud edge nodes that was demonstrated at Empire Challenge 2011 using biometric services and mobile devices. He architected and developed the Fusion Exploitation Framework for the Army to provide the fusion framework for the DCGS-A platform. He has developed and fielded tactical airborne and ground-based SIGINT and MASINT data collection, mapping, and geo-location applications. He has written several papers on data processing infrastructures, including a paper presented at NSSDF in 2010 on Adaptive Sensor Processing Supporting Event-Driven Fusion.

Figure 2: Latent Relationship Discovery Platform



6 Cost Volume Itemized Listing (a-k)

a. Special Tooling and Test Equipment and Material

N/A

b. Direct Cost Materials

N/A

c. Other Direct Costs

N/A

d. Direct Labor

Name	Labor Category	Rate	Hours
Jeff Pirkey	Principal Investigator	$80/hour	964

e. Travel

Trip Purpose	Travelers	Airfare	Per Diem	Lodging	Destination
Kick-off Meeting	1	$500	$56	$87	Wright-Patterson Sire Force Base, Ohio
Continuation Kick-off Meeting	1	$500	$56	$87	Wright-Patterson Air Force Base, Ohio

f. Cost Sharing

N/A

g. Subcontracts

N/A

h. Consultants

N/A

i. Any exceptions to the model Phase I purchase order (P.O.)

N/A

j. DD Form 2345

N/A - The topic is not ITAR or EAR

k. Certifications

Attached

AIR FORCE SMALL BUSINESS INNOVATION RESEARCH (SBIR)/ SMALL BUSINESS TECHNOLOGY TRANSFER (STTR) PROGRAMS PHASE I AND II AWARD CERTIFICATIONS – PROPOSAL SUBMISSION

Small businesses submitting SBIR/STTR Phase I and II proposals must provide a completed copy of this document with the proposal package. All questions must be answered and an authorized officer of the company must sign and date it prior to submission.

The information included in this document, in combination with the SBIR/STTR Phase I or II proposal coversheet, will be used to determine a firm's eligibility for award under the SBIR/STTR Programs. Definitions for terms used in this document are set forth in the Small Business Act (13 CFR Part 121), the Small Business Administration (SBA) SBIR and STTR Policy Directives, and statutory/regulatory provisions referenced within those authorities. Please note, similar documents will be utilized to ensure continued compliance at other times during contract performance, if selected.

If the Government Contracting Officer believes a business does not meet the size eligibility requirements at the time of award, the CO must file a size protest with the SBA. The SBA will then accomplish a size status determination with regard to eligibility. SBA may request supporting documentation and further clarifying information in the conduct of this determination. If, after award, the CO believes the firm does not meet eligibility requirements, the CO may request supporting documentation and clarifying information to assist in verification of contractor information already received.

No information already provided to the Federal Government affects its right to pursue criminal, civil, or administrative remedies for incorrect or incomplete information provided in this document or on the proposal coversheet. Signatory authorities for this document may be prosecuted if false information is provided.

The undersigned has reviewed, verified, and certifies:

1) The business concern meets the ownership and control requirements set forth in 13 CFR 121.702.
✓ Yes ____ No

2) If a corporation, all corporate documents (articles of incorporation and any amendments, articles of conversion, bylaws and amendments, shareholder meeting minutes showing director OR officer elections, organizational meeting minutes, all issued stock certificates, stock ledger, buy-sell agreements, stock transfer agreements, voting agreements, and documents related to stock options, including the right to convert to non-voting stock or debentures into voting stock) evidences meeting the ownership and control requirements set forth in 13 CFR 121.702.
✓ Yes ____ No ✓ N/A Explain: Not a corporation

3) If a partnership, the partnership agreement evidences meeting the ownership and control requirements set forth in 13 CFR 121.702.
____ Yes ____ No ✓ N/A Explain: Not a partnership

4) If a limited liability company, the articles of organization and any amendments, and operating agreement and amendments, evidences meeting the ownership and control requirements set forth in 13 CFR 121.702.

✓ Yes ____ No ____ N/A Explain: _____

5) The birth certificate, naturalization papers, or passports demonstrate any individual relied upon to meet eligibility requirements are U.S. citizens or permanent resident aliens of the United States.

✓ Yes ____ No ____ N/A Explain: _____

6) There is no SBA size status determination currently in effect finding the firm exceeds the 500 employee limitation, including employees of its affiliates.

✓ Yes ____ No ____ N/A Explain: _____

7) If selected, the awarded research/research and development (R/R&D) will be performed in the United States; or a deviation will be approved in writing by the Government Contracting Officer.

✓ Yes ____ No ____ N/A Explain: _____

8) If selected, performance will take place in the firm's facilities with the firm's employees, except as otherwise identified in the proposal, agreed to during contract negotiations, and approved through bi-lateral signature of the resulting contract.

✓ Yes ____ No ____ N/A Explain: _____

9) If applicable, the firm has registered in the SBA database as majority-owned by a venture capital operating company (VCOC), hedge fund, or private equity firms. (**NOTE**: IAW Section 4.4 of the DoD SBIR Solicitation, firms owned by multiple VCOCs, hedge fund, or private equity firms are INELIGIBLE to submit proposals unless the aggregate of all owners, affiliates, subsidiaries, and other firms in which they invest meets the requirements of a small business concern as found in 13 CFR 121.702.)

____ Yes ____ No ✓ N/A Explain: Not applicable

10) The firm will notify the Air Force immediately if all or a portion of the proposal work is funded by another Federal agency. ✓ Yes

11) The firm understands information submitted may be provided to Federal, State, and/or local agencies to be used for determining violations of law and other purposes. ✓ Yes

The undersigned is an officer of the business concern and authorized to represent and sign this certification on its behalf. By signing this certification, the undersigned certifies the information provided in this document and the proposal coversheet, as well as all information within the proposal, is true and correct as of the date of proposal submission. The undersigned understands any intentional or negligent misrepresentation of the information contained in this document, the proposal coversheet, or any other part of the proposal may result in criminal, civil, or administrative sanctions, including but not limited to: fines, restitution, and/or imprisonment under 18 USC 1001; treble damages and civil

penalties under the False Claims Act, 31 USC 3729 et seq.; double damages and civil penalties under the Program Fraud Civil Remedies Act, 31 USC 3801 et seq.; civil recovery of award funds; suspension and/or debarment from all Federal procurement and non-procurement transactions, FAR Part 9.4 or 2 CFR Part 180; and other administrative penalties including termination of active SBIR/STTR awards.

Signature

1/20/2014
Date

JEFF T. PIRKEY
Printed Name (First, MI, Last)

PRESIDENT / CEO
Position Title

REALITY WARP SOFTWARE
Company Name