

Standardizing Event Rule Languages



Alternative Title: Do We Really Need Another Rule Language?



Paul Cichonski (NIST)
George Saylor (G2)



Goals of Common Event Rule Expression (CERE)

- Enable EMAP rule authoring and rule migration and sharing activities.
- Provide vendors and consumers a way to express and share rules for correlation, filtering and aggregation of event data.
 - Use existing event data vocabularies to enable the assertion of specific relationships when certain data patterns exist (e.g., an incident is occurring if the following events are seen).
 - As a **rule interchange format**, it does not have to be executable, but must allow translation down to executable languages.

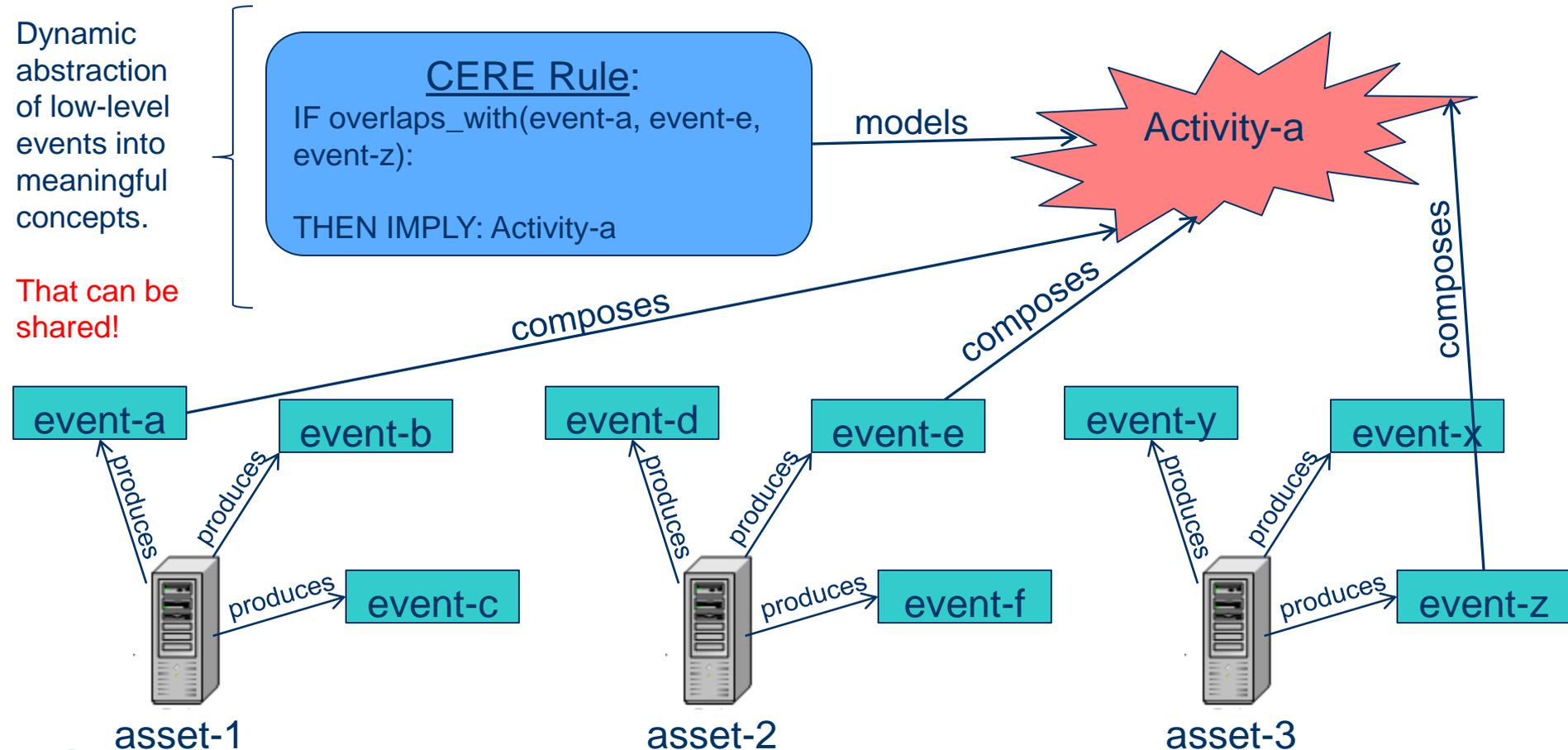


Goal of this Meeting

- **Brief Ideas:** present possible strategies for building a standardized rule language for correlation, aggregation, and filtering rules within event management.
- **Solicit Feedback and Requirements:** We are mainly looking for feedback on which strategy we should pursue; no concrete solutions exist yet, just ideas on possible paths to follow.



Standardized correlation rules will enable sharing of ongoing multi-event activity → If we do it right!





Rules are the other half of enabling the automation of “situational awareness” (SAW) within event/log management

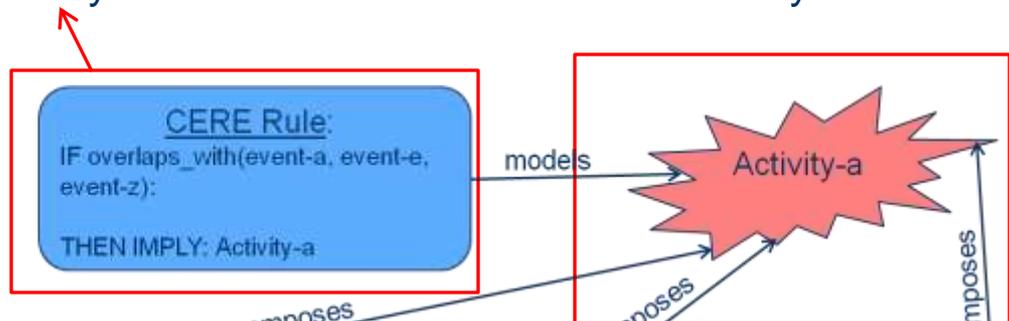
- *“The domain knowledge that is required for SAW [or event management] is of two types:*
 - *1) knowledge about what classes or objects, attributes and relations are possibly relevant and*
 - *2) what conditions must exist among the objects and their attributes for a given relation to hold true¹.”*
- The first component is the granular vocabulary of the domain (e.g., CEE).
- The second component is the need to use the granular vocabulary to draw conditional relationships/inferences and connect with a higher-level vocabulary → CERE.

¹ Matheus, Kokar, Baclawski, Letkowski. Constructing RuleML-Based Domain Theories on top of OWL Ontologies 08/30/2011

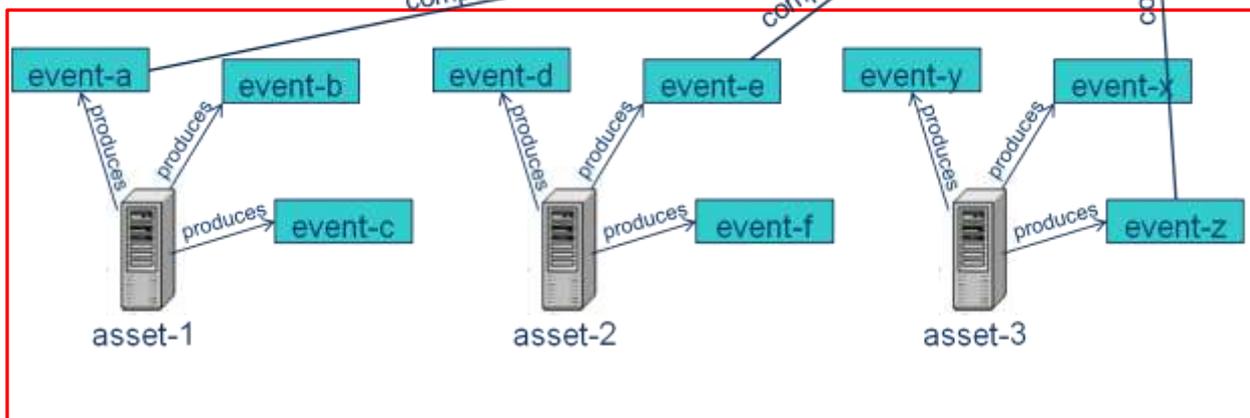


Different type of vocabularies (or models) are connected through rules

Language for dynamically describing how instances from event vocabulary combine to form instances in activity vocabulary.



Vocabulary describing how composite-events form specific activities, observables, or incidents.



Low-level vocabulary for describing events and event properties.



A Very Simple Classification of Rule Types*

- **Logic or Inference Rule**: A rule that results in the *assertion of new facts* if a specific condition is met.
 - Event correlation rules fall into this classification.
 - For example: if (event1, event 2, and event 3 occur at the same time) then infer that (activity x is occurring).
- **Action Rule**: A rule that results in the *modification of existing data* if specific conditions are met.
 - Event filtering and aggregation rules fall into this classification.
 - For example: if (event 1 record contains source_ip), then modify (event 1 record to replace source_ip with fake data).
- Similar syntax and vocabulary for both, but the results differ.

* Slightly higher level version of RIF classification
(<http://www.w3.org/TR/rif-overview/>) to meet the needs of this discussion.



A little more history on “rules”

“Rule-languages and rule-based systems have played seminal roles in the history of computer science and the evolution of information technology. From expert systems to deductive databases, the theory and practice of automating inference based on symbolic representations has had a rich history and continues to be a key technology driver. ... Rules themselves represent a valuable form of information for which there is not yet a standard interchange format, although significant progress has been made within the RuleML Initiative and elsewhere.”¹

- **What This Means:** This is an old problem with different solutions; that means we have multiple options and sub-options. One of these options is building our own, but it may not be the best one.

¹RIF UCR (<http://www.w3.org/TR/rif-ucr/>)

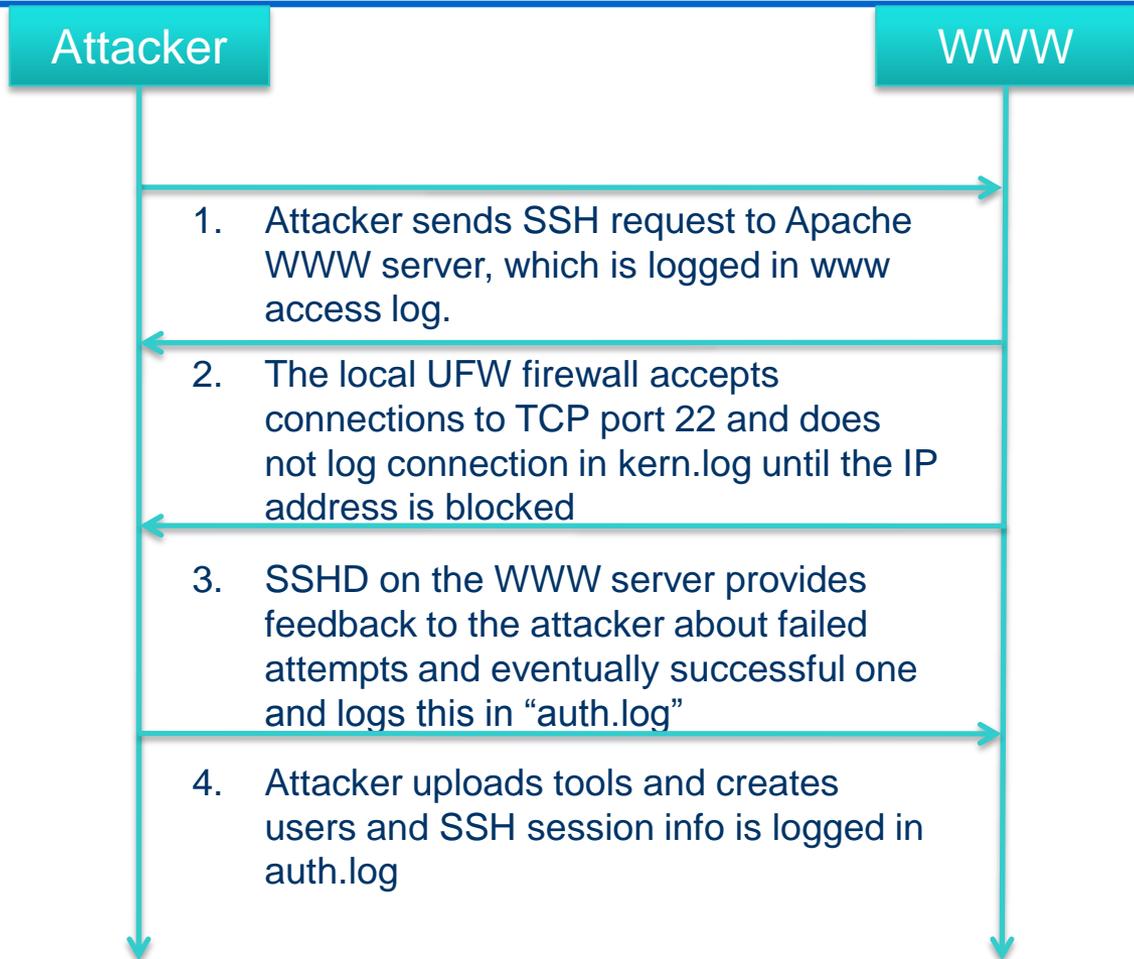


Possible Options – No detailed solution exists for any option; we need to decide which to pursue.

- **Option 1**: Adopt an existing rule interchange format.
- **Option 2**: Build our own from scratch, or evolve an existing language to include support for CEE.



Example Scenario Involving a Malicious Incident → Use this to help explore options



Log Source: honeynet challenge
(http://honeynet.org/challenges/2010_5_log_mysteries)



Questions that model the composite activity

- If a single internet address attempts to login to n user accounts and fails, it may imply a *brute force* attack.
- If an internet address that is implicated in a potential brute force attack is later demonstrated to successfully login, it may imply evidence of *compromise*.



OPTION 1

Adopt an existing rule interchange format.



Rule Interchange Formats are about exchange, not execution

- Execution of the rule will normally happen after it has been translated into an executable language.
 - Parallels our data model work; data models are about exchange, not implementation.
 - Rule Interchange Formats allow for rules to be expressed in domain specific vocabularies.
- Two leading efforts in rule interchange:
 - W3C Rule Interchange Format (RIF)
 - The Rule Markup Initiative (RuleML)
- Choice now is not between RIF and RuleML, but if we should pursue this option (and eventually pick a specific interchange format)



Rule Interchange Format

- W3C Rule Interchange Format (RIF)
 - Designed for the purpose of exchanging rules
 - Reasonable momentum as a standard (accepted as a recommendation by W3C)
 - Is highly expressive and extensible, but also generic (maybe too generic)
- Multiple dialects exist for different types of rules
 - RIF-BLD for logic-based rules
 - RIF-PRD for action rules
 - RIF-Core is a subset of both RIF-BLD and RIF-PRD and allows users to create rules that may be processed by both types of rule engines.



Expressing “Brute Force” question in RIF Core syntax

```
Prefix(attack <http://scap.nist.gov/attacks#>)
Prefix(cee <http://cee.mitre.org/>)
```

Declare namespaces where vocabularies are defined.

```
Forall ?event ?src_ip (
  attack:brute_force(?src_ip) :-
    AND(cee:failed_login(?event 5 60)
        cee:src_ipv4(?event ?src_ip)))
```

Define variables to be used in rule.

Head of rule (or the “THEN” portion). This is implied if the portion of the rule after the ‘:-’ evaluates to true. The “attack:brute_force” is a unary predicate asserting that the ?src_ip is responsible for a brute_force attack.

Body of rule (or “IF” section). cee:failed_login is a ternary predicate stating multiple ?event(s) had 5 failed logins in 60 minutes and that event is associated with ?src_ip.

- RIF provides a framework for connecting vocabularies with logical implications, nothing more.
- This assumes ‘cee:failed_login’ relationship is a CEE actionTag and cee:src_ipv4 is a field relationship.
 - CERE spec would have to define how to interpret CEE terms as predicates of a rule (different types of terms may be interpreted differently – binary vs ternary).
 - This is only one possible way of doing it; alternatively, predicates could be defined that make use of CEE terms.
- XML syntax also available.
- **CERE will need to specify everything relating to how our vocabularies are used within RIF.**



Alternative way to express “Brute Force” rule in RIF Core syntax

```
Prefix(attack <http://scap.nist.gov/attacks#>)
```

```
Prefix(cee <http://cee.mitre.org/>)
```

```
Prefix(tmprl <http://nist.gov/temporal_relationships#>)
```

```
Forall ?event ?src_ip (
```

```
  attack:brute force(?src_ip) :-
```

```
    AND(tmprl:within(
```

```
      cee:term(?event cee:action "failed_login")
```

```
      5
```

```
      60)
```

```
      cee:term(?event cee:src_ipv4 ?src_ip))
```

Some vocabulary of temporal relationships useful in correlation.

tmprl:within is a pre-defined temporal predicate. The first argument is an event that happened, the second argument is the amount of times it happened, the third argument is the timeframe. A Rule engine would evaluate this against a KR to determine if available facts meet this condition.

A ternary predicate that relates an arbitrary event to some object via a CEE field/tag.

- An alternative way of expressing previous rule.
 - Provides mechanism for asking questions about CEE events without the need to further specify fields/tags.
 - Provides simple mechanism to relate an event to an object through a pre-defined CEE field name.



Simplified RuleML version of “Brute Force” question

```
<implies>
  <head>
    <Atom>
      <Rel>attack:brute_force</Rel>
      <Var>ip_address</Var>
    </Atom>
  </head>
  <body>
    <Atom>
      <Rel>cee:failed_login</Rel>
      <Var>ip_address</Var>
      <Ind>5</Ind>
      <Ind>20</Ind>
    </Atom>
  </body>
</implies>
```

- Fairly similar concept with slightly different syntax and semantics behind rule concepts
 - Allows the same type of vocabulary integration
 - More research is required to determine which exchange language is better suited to our problem.



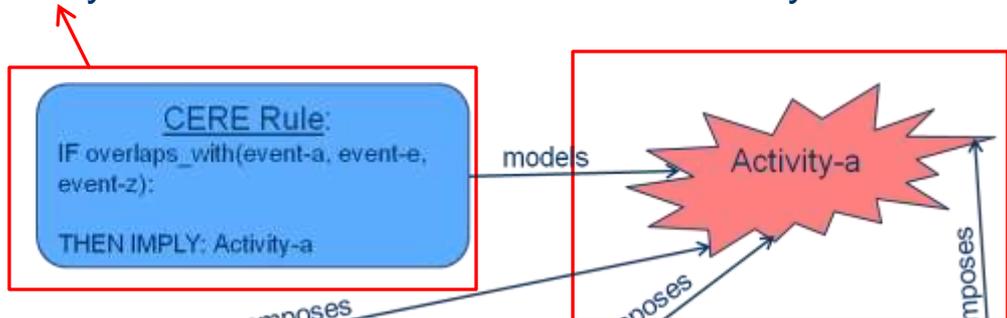
Question to consider throughout

- Should we standardize on CEE (or some other event model) only, or should we try to allow plug-and-play between multiple disparate event data models?
 - Core difference here is: are we trying to make the solution vocabulary-agnostic or specific?
 - The more disparate vocabularies we support, the longer and more complex the spec.
- XML-based vocabularies make it harder to create general solutions higher up in the stack.
 - Syntax differences between models require different methods of integrating with a rule exchange model like RIF.
 - CEE is easier than others due to the use of profiles defined outside of the core schema.
- This decision will drastically change the scope of the problem and the complexity of the solution.

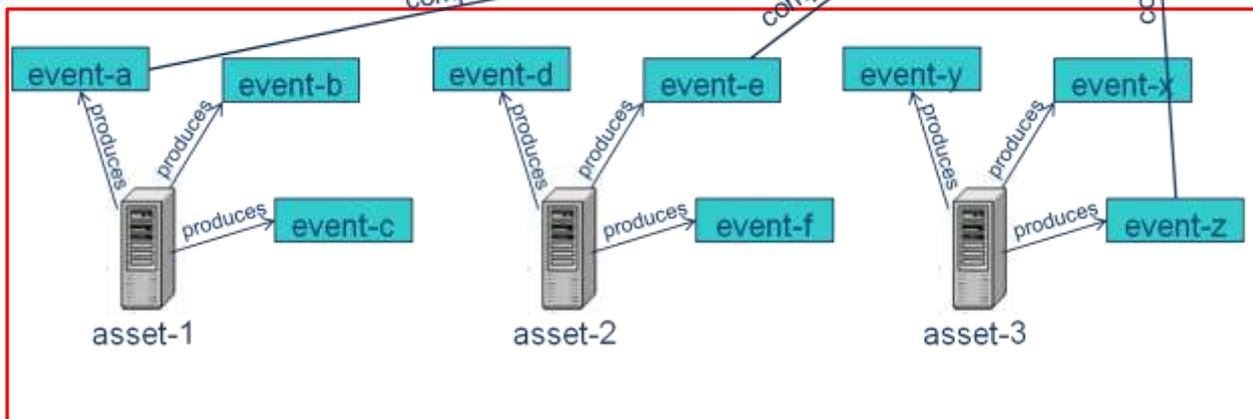


The more vocabularies we support, the harder this [standardized] integration becomes.

Language for dynamically describing how instances from event vocabulary combine to form instances in activity vocabulary.



Vocabulary describing how composite-events form specific activities, observables, or incidents.



Low-level vocabulary for describing events and event properties.



CERE Spec for option 1 will focus on vocabulary integration with rule exchange.

- How do content creators use event data vocabulary within the body of a rule?
 - This is hard when we are using XML schema-based approach (RDF would be easier, but...)
 - This becomes much easier if we can agree on a single XML vocabulary.
- Vocabulary to use for describing what a rule implies
 - How to describe the composite-activities, incidents and observables?
 - This does not have to happen immediately.
- How to define new relationships that may be used in these rules?
 - Temporal Logic: Allen's Interval Algebra
 - Others?



OPTION 2

Build our own rule language from scratch, or evolve an existing language to include support for CEE.



Building a rule language is a complex task.

- There are many drawbacks:
 - Essentially duplicates some of the work performed elsewhere and competes in an already crowded marketplace.
 - Adoption could be challenging if the language is too domain specific.
- This option would give us the most flexibility.
 - Can be purpose-built for the log management domain and the standardized event data model of our choice.
 - Would be under full control of the group.
- We should avoid this option if possible.



Back to an important question

- Should we standardize on CEE (or some other event model) only, or should we try to allow plug-and-play between multiple disparate event data models?
 - Core difference here is: are we trying to make the solution vocabulary-agnostic or specific?
 - The more disparate vocabularies we support, the longer and more complex the spec.
- XML-based vocabularies make it harder to create general solutions higher up in the stack.
 - Syntax differences between models require different methods of integrating with a rule exchange model like RIF.
 - CEE is easier than others due to the use of profiles defined outside of the core schema.
- This decision will drastically change the scope of the problem and the complexity of the solution.



How to enable rule tagging?

- Tag rules for logical grouping so you can select/unselect certain rules based on what they do.
 - Can also use tags in a pre-processor to determine if rule needs to be run against a specific data set.
 - This would require an extensive metadata language about rules.
 - This would essentially be a separate spec.
- Is this needed? Will anything else within security automation work?



What are the requirements for making future decisions?

- Expressiveness/completeness of data model?
- Efficiency of data model when translated to executable code?
- Required support for specific features (e.g., ability to translate all CERE rules into native code)?
- Extensibility of the data model?
- Modularity of the data model (e.g., some tools may not want to support all translation types)?



Questions & Answers / Discussion



Paul Cichonski

National Institute of Standards and
Technology (NIST)

paul.cichonski@nist.gov

(301) 975-5259



EXTRA