

## Probe description DB (ProDesDB)

As per the discussion with James on 13<sup>th</sup> Jan 2009 I came to know about following things that need to be included in the ProDesDB.

In the transition phase from SAM to Nagios infrastructure for scheduling tests, some development had been done with hard coding the test and probes descriptions. These probes are then aggregated to define a service.

Now the idea of ProDesDB is to remove these hard coding and store this information in a DB and provide interface for nagios and other applications to use the stored information for scheduling probes.

ProDesDB would include:

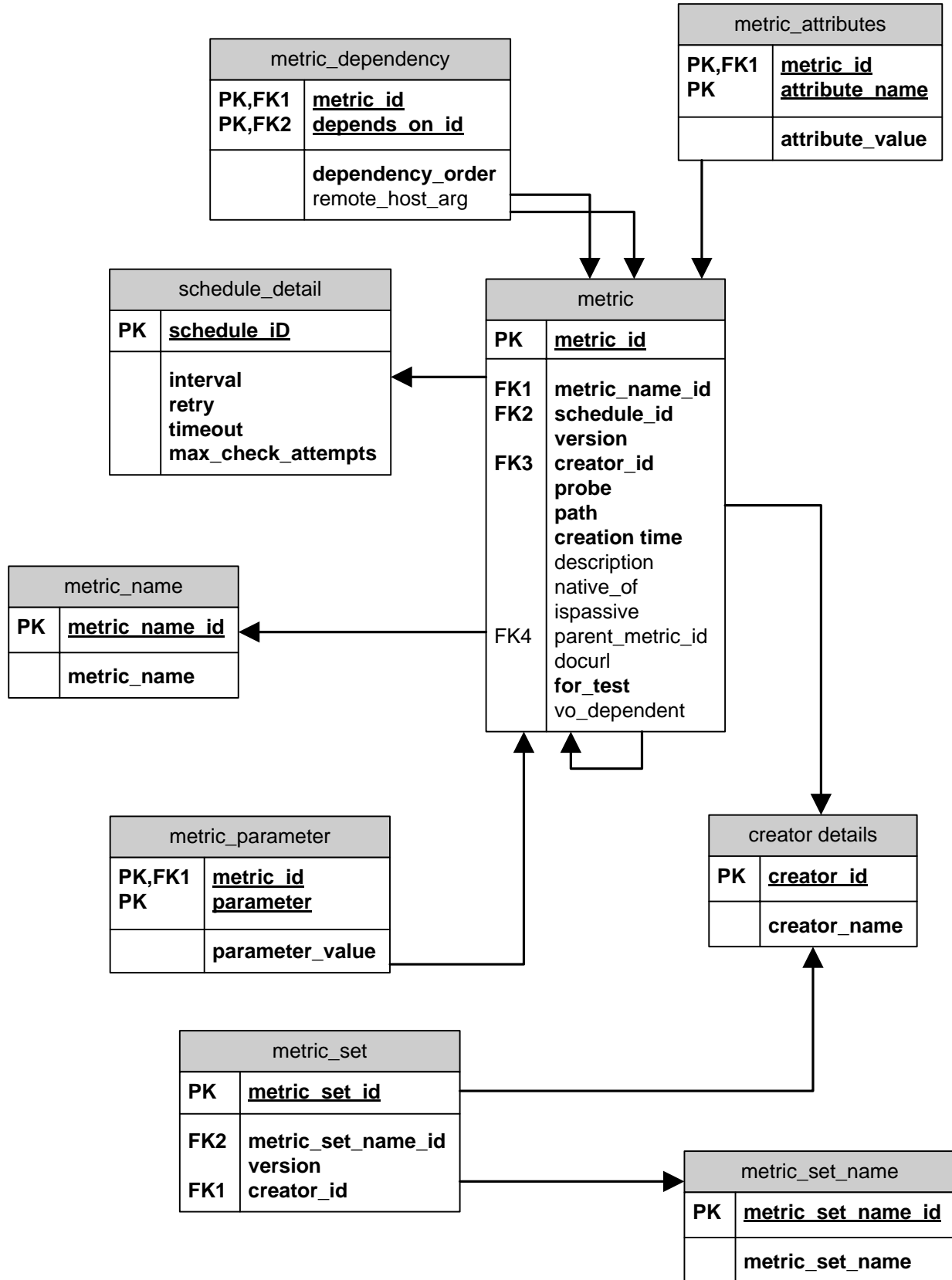
1. Description of metric (description, name, probe, docurl, attributes, path)
2. Scheduling information of Metric (interval, retry)
3. Dependency of metrics on other metrics (Comment on whether order of dependency is also required?)
4. Metadata of the metric (version, creator, submitter etc)
5. Definition of Metricsets (metrics can be grouped together in the form of metricset)
6. Description and metadata for metricset
7. Servicetype defines other level of grouping it is defined by a set of a metricset (comment on whether direct inclusion of metric in a service type should be included?)
8. Description and metadata of servicetype
9. Site and node groups expressions : these are required to define different subset of node to be scheduled on which a servicetype should be scheduled
10. History of the metric, metricset, servicetype and group expression is to maintained: Required to get information of the probe description at a time in the past
11. Interface for adding, deleting, editing these elements of ProDesDB
12. Interface for querying the probe description and schedule

Things that are not included ProDesDB are:

1. Storage of test results.
2. Details of site and node information.
3. Availability calculation algorithms from the test results.

Based on these assumptions a simple Data model is created with ERD in ProDes1.vsd. Please refer to it.

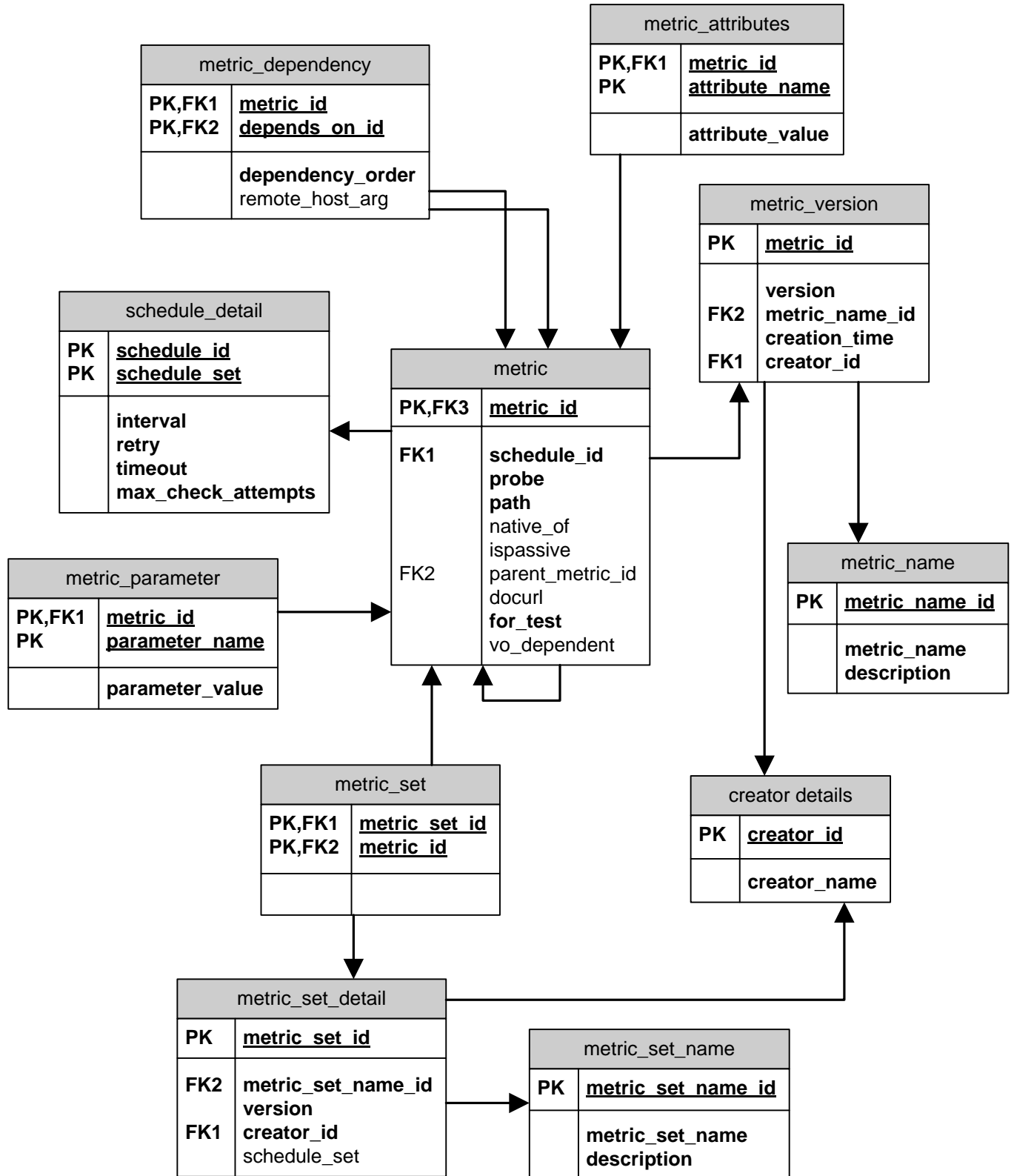
# Probe Description DB Data Model 1



Due to some short comings changes are made to these data model. New ERD is available as ProDes2.vsd. Enhancements to this model are:

- A new metric\_id is given to different version of a metric. This will help a metric set to use an old version of a metric if it wants.
- Description of a metric resides in meric\_name table since for a metric name its description remains same. There can be multiple metric\_id with same metric\_name (different version), so the metric\_name is kept in a separate table.
- Creation\_date and creator\_detail of metric only changes with version so it is kept in a common table metric\_version.
- All the other tables that contain metric\_id as FK refer to metric table, not metric\_detail table since it is only available once its schedule and other information is available.
- Concept of schedule\_set is introduced to allow a metric set to have a custom schedule for a metric. This will allow different metric\_set to differently schedule metric(s). If schedule\_set of a metric set is not defined or there is no schedule\_set, schedule\_id pair is available, then schedule\_id of schedule\_set '0' is selected. \*\*\*\*\*
- Correct place of docurl field of metric table need to be reviewed. \*\*\*\*\*

# Probe Description DB Data Model 2



Models drafted up to now don't have history information. To include history information in the model there 2 approaches were thought, these are:

- A. To keep the history of state of DB itself; another table with same fields and some extra field is created and operations are logged. This is the way Joshi Sir had suggested, and probably uses in Topology DB.
- B. To keep the history state of information contained in DB; some additional fields are incorporated in the table itself. Both the past and current information remains in the same table.

I suggest is to use option 'B' since it suits our requirement (see below). Moreover it had an advantage over 'A'; queries for extraction of current and past information are same and refer to same table.

For history 3\*\*\*\*\* fields are being added to these tables. Their semantics are as follows:

- Available\_after: Since in ProDesDB probes and other information should be loaded much before they are actually used by the Nagios. So this field defines start timestamp after which a particular record is considered to be exposed to outside.
- Deleted\_after: Giving reason similar to above we can continue to have a timestamp field for deleted instead of isDeleted flag.
- Not\_available\_after: There is a slight difference between deleted\_after and not\_available\_after fields. First one is used when we try to invalidate a record after a timestamp. It will not be exposed even if the higher level groups reference them, but in later case they are available for higher level groups that already reference it, but new one could not refer them.

Records from these tables will never be deleted. Update to records of these tables will result into 2 steps:

- a) Update of old's not\_available\_after and delete\_after fields
- b) Insert of a new record, duplicate of old, with new available\_after field

This will require including available\_after field in primary key of all these tables. I am little bit confused with use of 2 fields not\_available\_after and deleted\_after. For the time being I have created them but, after the design and implementation of upper layers one of it might not be required and dropped.

Changes to ProDes2.vsd are made with following assumptions:

- We are not going to have Delete and update operation on the following tables.
  - metric\_set\_name
  - metric\_name
  - metric\_version
  - metric\_set\_detail
- Creator\_details have much significance it just identifies the creator uniquely. Giving it a low significance in the ProDesDB, for the time being its history information is not modeled.
- For the rest of the tables 3 fields for history are going to be included. All these tables will have available\_after field in their PK

Augmenting with the above feature the new ProDesDB model will look like as shown in ProDes3.vsd

# Probe Description DB Data Model 3

