

InFlux DB schema (E Release)

This page documents the inFlux DB schema used in the E Release for the KPM (and other) information collected from E2 Simulator.

Version	Author	Comments	Date
0.1	Agustin F. Pozuelo	Initial DRAFT, calling for comment!	24 Sep 2021



DRAFT / Proposal

This schema is open for discussion and contribution from the O-RAN community

Summary / Goals

The goal is to have a generic schema definition for xApps to access timeseries data obtained via E2SM/KPM and possibly the other E2SM protocols.

References

- E2SM-KPM specs
- TS 28.552 "5G performance measurements"
- InfluxDB glossary - <https://docs.influxdata.com/influxdb/v1.8/concepts/glossary>
- Viavi RIC Test User Manual

Glossary

- **SDL**: Shared Data Layer, a term referring to the services and schemas used to share data among xApps

Essential InfluxDB terms:

- **Database**: the highest level unit of storage
- **Measurement**: a **table** inside a database
- **Data point**: a **row** inside a Measurement table.
- **Field**: a sort of **column** of a Measurement table, with name and type of data.
 - *Note that not all fields need to be present in all data points*
- **Tag**: a **metadata field** that is indexed so queries on them are performant

Guidelines

- For items not explicitly defined in O-RAN specs, look for relevant 3GPP specs.
- Field and tag names should be brief because they may pose a direct impact on InfluxDB performance at high loads.
- For proprietary/enrichment data fields, use a brief vendor prefix e.g. "Viavi."

Proposed Schema for E2SM/KPM data

- Database name: "**E2SM-KPM**"

For the labelling of fields we will be using names from E2SM/KPM v2.0 (MeasurementLabel) and TS 28.552 "5G performance measurements" as much as possible.

Following the general ideas in from TS 28.552, using short labels, organized in hierarchical form <topic>[.<subtopic>].<name>

KPM Measurement tables:

1. "**CellReports**" table containing the following fields:

- a. **Timestamp**
- b. **PLMN** (*tag*)
- c. **KPMNodeID** (*tag*) - The KPM Node identification in string form to accommodate the many binary variants (maybe an hex dump?)
- d. **S-NSSAI** (*tag?*) - useful to identify slices?
- e. **NRCellIdentity** (*tag?*)

Any of the fields defined for cell reports in TS 28.552, for example the following are provided by RIC Test scenario generator:

- f. **DRB.UETHpDI** - Average DL throughput in Kb/s
- g. **RRU.PrbUsedDI / RRU.PrbAvailDI**
- h. **RRU.PrbUsedUI / RRU.PrbAvailUI**
- i. **QosFlow.PdcpPduVolumeUI** - UL PDCP PDU Data Volume (amount of kilobytes in the reporting period)

- j. **QosFlow.PdcpPduVolumeDI** - DL PDCP PDU Data Volume (amending inconsistency in the case of the final "L" in TS 28.552)

And proprietary extensions:

- k. **Viavi.Geo.x** - Non standard / Viavi proprietary extension / External enrichment
- l. **Viavi.Geo.y**
- m. **Viavi.Geo.z**

- 2. **"UeReports"** measurement table containing the following fields:

- a. **Timestamp**
- b. **UE.Id** - String representation to accomodate the many forms.
Note: A "UE id resolution" xApp may be needed to match the same UE on different measurements and possibly producing unique ids for this field
- c. **Slice.Id**
- d. **DRB.UETHpDI** - TS 28.552 Average DL UE throughput
- e. **RRU.PrbUsedDI** - TS 28.552 DL PRB used for data traffic
- f. **Viavi.Geo.x**, **.y**, **.z...** Viavi proprietary extensions
- g. **RF.serving.Id** - May be just the PCI or resolved into NRCellIdentity
- h. **RF.serving.RSRP** (Loosely based on LTE specs)
- i. RF.serving.RSRQ
- j. RF.serving.RSSINR
- k. RF.nb1.CellId - 1st neighbour id
- l. RF.nb1.RSRP, etc. 1st neighbour signal levels
- m. RF.nb2.CellId - 2nd neighbour id
- n. RF.nb2.RSRP, etc. 2nd neighbour signal levels

- 3. **"cellMeasReport"** cell load measurement table has the following fields:

- a. **Timestamp**
- b. **DLOccupyPRBNum** – 3GPP supported
- c. **CellIDLMACRate** – proprietary
- d. **ULSINR** – proprietary
- e. **MCS** – proprietary
- f. **PDCPOccupBuffer** – proprietary
- g. **PDCPUnusedBuffer** – proprietary
- h. **DLPacketDiscardNum** – proprietary
- i. **DLPacketSDUNum** – 3GPP supported
- j. **DLPacketLossNum** – 3GPP supported
- k. **DLMACRate** – proprietary