Report for the Next Generation Data Format

Registered Voters

Voting registry: https://github.com/FDSN/miniSEED3-TechnicalEvaluation/issues/1

10 FDSN institutions registered to vote represented by the following people:

GitHub Handle	Name	Institution
@ketchum-usgs	David Ketchum	ANSS-USGS
@ozym	Mark Chadwick	GNS Science
@wtliang	Wen-Tzong Liang	Institute of Earth Sciences, Academia Sinica, Taiwan
@chad-iris	Chad Trabant	IRIS DMC
@crotwell	Philip Crotwell	South Carolina Seismic Network
@ihenson-bsl	Ivan Henson	Berkeley Seismological Laboratory, UC Berkeley
@kaestli	Philipp Kästli	SED-ETHZ, (Swiss Seismological Service)
@JoseAntonioJara	Jose Antonio Jara	ICGC, (Cartographic and Geological Institute of Catalonia), Spain
@ValleeMartin	Martin Vallée	GEOSCOPE/IPGP
@claudiodsf	Claudio Satriano	RESIF (French seismologic and geodetic network), France

Voting Results

Requirement: A new method of identifying a time series

https://github.com/FDSN/miniSEED3-TechnicalEvaluation/issues/4

Summary

There seems to be consensus to using a single time series identifier in the approximate form of `FDSN:<network>_<station>_<location>_<channel>`. Details to what each of these mean (and if all 4 or even more are needed) are discussed in #27-#30. This issue is purely about using a single but very flexible namespaced string identifier.

Questions

1. Are you in favor of adapting a single string based and namespaced time series identifier? (Yes/No)
   - **8 x YES | 0 X NO**
2. Should the namespace field be mandatory? (Yes/No - otherwise it would default to the `FDNS:` namespace).
   - **8 x YES | 0 X NO**
3. What should the maximum length of the identifier be? (255 bytes/propose other)
   - **7 x 255 bytes | 1 x 64k**
4. What should the text encoding be? (UTF-8/ASCII/propose other)
   - **7 x ASCII | 1 x UTF8**
   - Comment by @chad-iris: The ASCII subset used for SEED 2.4 plus a few extra characters already proposed.
   - Comment by @kaestli: ASCII with %-Escaping (follow chapter 2 of W3C RFC 3986 on URI - see https://tools.ietf.org/html/rfc3986#section-2.1)
   - Comment by @crotwell: utf8 - but I think given the character limitations in #27-#30 this is effectively the same as ascii but may give some future flexibility for the future. I would also say defer to existing standards wherever there is conflict, ie rfc3986.
   - Comment by @claudiodsf: ASCII, since the identifier will have fixed length and accents are mostly a problem in IDs. But, what happens for non-latin alphabets?

Requirement: The new format must be fully self-defining

https://github.com/FDSN/miniSEED3-TechnicalEvaluation/issues/5

Summary

This requirement is a bit undefined and not much discussion happened.

Questions

1. Should the new data format be parse-able without resolving to heuristic checks? E.g. a mandatory byte order field. (Yes/No)
   - **9 x YES | 0 X NO**
2. Should information not in the fixed header (assuming we have one) be in some form of standardized format (e.g. no binary blockette that is meaningless without the spec)? (Yes [some standardized format] / No [keep binary blockette layout or something similar])
   - **7 x YES | 2 X NO**
   - Comment by @ketchum-usgs: No - No undefined add ons in the headers
   - Comment by @chad-iris: Yes, with FDSN defining "reserved" headers.
   - Comment by @kaestli: No - no undefined add ons in the headers.
     - Reply by @chad-iris: The description of this item does not include "undefined add ons". I think it is about how optional headers are structured.
   - Comment by @ozym: Yes, assuming some form of namespacing.

   - Comment by @claudiodsf: Yes. This will provide a flexible format, with still the possibility of having specific header add ons for a certain community (e.g., FDSN extensions)
3. Assuming 2. results in a “yes”: What should this format be? Please propose one or more.
   - JSON
   - JSON5
   - CBOR
   - MessagePack
   - BSON
   - UBJSON
   - In general there is preference for a binary format.

Requirement: Specify absolute time, including leap seconds, using an existing standard

https://github.com/FDSN/miniSEED3-TechnicalEvaluation/issues/6

Summary

Seems to very clear cut - a format that can represent leap-seconds is mandatory in any case so we are voting here for the time representation. There are two choices (both with nanosecond precision):

Questions

1. The binary struct introduced by @chad-iris above:

```
Year (0 - 65535)                    UINT16
Day-of-year  (1 - 366)              UINT16
Hour (0 - 23)                       UINT8
Minute (0 - 59)                     UINT8
Second (0 - 60)                     UINT8
Nanosecond (0 - 999999999)          UINT32
```

2. An ISO 8601 based string representation in some variant, e.g. something akin to `2018-01-11T11:26:52.829308846Z`.

Please vote on which you would prefer. (Binary struct / ISO 8601 string)

- **7 x binary struct | 0 X ISO 8601 string**
- Comment by @kaestli: Binary struct, however with year=int32 (for planetary modelling)

Requirement: Conversion tools must perform conversion from MiniSEED 2 without loss of important information

https://github.com/FDSN/miniSEED3-TechnicalEvaluation/issues/7

Questions

1. Is data-loss acceptable at all? (Yes/No) - all other points will not be valid anymore if this is a “no”.
   - **8 x YES | 0 X NO**
2. Is it acceptable to loose information regarding the *Beam related blockettes (400 & 405)*? (Yes/No)
   - **8 x YES | 0 X NO**
3. Is it acceptable to loose information for *Blockette 2000*? (Yes/No)
   - **8 x YES | 0 X NO**
4. Is it acceptable to loose different “clock model” designations if multiple blockette 500s are in a single record? (Yes/No)
   - **8 x YES | 0 X NO**
5. Is it acceptable if the “Sequence number” field in miniSEED 2.x is only optionally retained? (Yes/No)
   - **8 x YES | 0 X NO**
6. Is it acceptable if the “Data header & quality indicator” field in miniSEED 2.x is only optionally retained? (Yes/No)
   - **6 x YES | 2 X NO**
   - Comment from @claudiodsf: No. It should be mandatory. Moving this to metadata would produce too much complexity (a new metadata period every time the quality flag changes)
     - Reply by @chad-iris: To clarify, I do not believe this implied moving the D, R, Q, M value to metadata, but instead to make it optional within the time series data format.

Requirement: Joint evolution of StationXML

https://github.com/FDSN/miniSEED3-TechnicalEvaluation/issues/8

Summary

No vote as the unanimous consensus is that only moderate changes to StationXML will be required (if any at all). Any incompatible changes in the new data format must force changes to the meta-data format in any case so this is not something that must be discussed but it must rather be kept in mind.

Requirement: Proper documentation, best practices

https://github.com/FDSN/miniSEED3-TechnicalEvaluation/issues/9

Summary

No vote as the unanimous consensus is that this should be done.

Requirement: Identification of non-raw, derived data

https://github.com/FDSN/miniSEED3-TechnicalEvaluation/issues/10

Summary

This is a bit of a complicated issue. I think we agree that full and proper provenance is not in the scope of the next generation data format and must be delegated to the meta data in some form. Also where exactly this information should go in the format is not clear and there are a large number of possibilities.

Question

Should there be a simple way to flag time series in the new format as either "raw" (whatever the exact definition of that is), "derived" (not "raw"), or "synthetic" (not based on actual recordings)? (Yes/No)

- **6 x YES | 2 X NO**
- Comment by @kaestli: No (not as a flag, as terms are not defined and overlapping.). But such streams should have different streamIDs and different Metadata
- Comment by @claudiodsf: Yes, but not a single flag, since the three definitions can overlap.
- Comment by @ValleeMartin: Yes but taking into account that definitions can overlap
- Comment by @JoseAntonioJara: No, I think this feature should be specified together with the rest of channel's metadata.

Requirement: New data payload encoding types (general, opaque)

https://github.com/FDSN/miniSEED3-TechnicalEvaluation/issues/11

Questions

1. Retain data encoding specification system as in miniSEED 2.x. (Yes/No)
   - **8 x YES | 0 X NO**
   - Comment by @chad-iris: Yes, with a number of encodings (e.g. DWWSSN) marked as deprecated.
   - Comment by @claudiodsf: Yes. With marking obsolete encodings as deprecated
2. Allow for an easy integration of additional data encodings without changes to the core definition. (Yes/No).
   - **8 x YES | 0 X NO**
3. Actively investigate alternative encodings. (Urgent/Not Urgent)
   - Yes, but no not urgent by general agreement.
   - Comment by @ozym: not urgent but would allow taking good advantage of variable length blocks if they are voted in.
4. Explicitly allow an "opaque" data encoding type. (Yes/No)
   - **8 x YES | 0 X NO**
5. Clearly state that any opaque data should not be exported by data centers and should be considered a transient transport mechanism in contained scenarios. (Yes/No)
   - **8 x YES | 0 X NO**

Requirement: Include a CRC (cyclic redundancy check) of the complete record

https://github.com/FDSN/miniSEED3-TechnicalEvaluation/issues/12

Summary

There is agreement that we want CRC but not what algorithm or what "type" of CRC. Technically it is also clear that the CRC field must be set to some pre-determined value (or ignored) for the actual calculation of the CRC.

Questions

1. What should the CRC include? (complete record / partial record / no record-level CRC)
   - **8 X "complete record"**
2. What algorithms should be used? (CRC-32/other suggestion)
   - **8 X CRC-32** - 2 people would also be okay with other lightweight algorithms.

Requirement: Include both a format and a data/publication version number

https://github.com/FDSN/miniSEED3-TechnicalEvaluation/issues/13

Questions

1. Do we want to include the actual data format version to enable self-identification and versioning of the data format? (Yes/No)
   - **8 x YES | 0 X NO**
2. Do we want a single byte "data publication version" somewhere in each record? This would be a linear version number without a lot of additional semantics largely useful internally for data centers. (Yes/No)
   - **7 x YES | 1 X NO**
   - Comment by @kaestli: No (a data stream which was modified should get a different streamID, not a different version number, but a streamID pretending it to be the same. What if "version" tag varies between records of the "same" stream?. Using the streamID for to point to metadata allows to further describe the version/modification there)
   - Comment by @ozym: Yes, although I could see this used as a mechanism to determine data providence within the collection systems (e.g. daisy chained data feeds) rather than version per se
   - Comment by @JoseAntonioJara: Yes, adding an identifier (namespace or other) meaning the datacenter where it was created.
3. Do we want a more complex "data publication version" which must include things like namespaces. (Yes/No)
   - **1 x YES | 7 X NO**
   - Comment by @crotwell: no, or at least not as a required header field. No objection to a standardized key that could be used in the optional part of the header as in #14
   - Comment by @kaestli: (this should be answered in the streamID discussion)

Requirement: Requirement: Allow addition of arbitrary headers by equipment manufacturers, operators, etc. without the need for authorization or approval by the FDSN

https://github.com/FDSN/miniSEED3-TechnicalEvaluation/issues/14

Questions

1. Do we want the ability to add arbitrary headers? (Yes/No)
   - **7 x YES** | **1 X NO**
   - Comment by @kaestli: No! A standard with non-standard (maybe opaque) extensions influencing the interpretation of the standardized information is not a standard.
   - Comment by @claudiodsf: Yes. This will allow a larger usage of the NGF by other communities. But we call for a standardization within the FDSN community.
   - Comment by ValleeMartin: Yes, but calling for a standardization within the FDSN community.
2. Assuming yes to 1. This either requires vendor defined definitions or some kind of key-value data storage format (of course with additional definitions but at least its parse-able without more information). Which do we want? (More definitions/key-value data format)
   - **7 x key-value** (one of them nested, might be implied by others) | **1 X NO**
   - Comment by @kaestli: XML with schema reference (or anything els as self-defining as possible). Implicit reference to the timespan of a record must be avoided by any means (as then, re-alignment of records would change the information content in an uncontrollable way.) key-value does not allow for this (at what time did key x have value y?)
3. Assuming we want a key-value data format. Which do we want? (Make a suggestion)
   - Deferred to "Requirement: The new format must be fully self-defining"

Requirement: Support variable length records for efficiency and flexibility

https://github.com/FDSN/miniSEED3-TechnicalEvaluation/issues/15

Questions

1. Do we want variable length headers without being restricted to a length of 2^X?
   - **8 x YES** | **0 X NO**
2. What should the maximum length be? Likely determined by the data type used to store the maximum length. (Uint16=65536 bytes / Uint32=4294967296 bytes / other suggestion).
   - **5 x Uint32** | **3 X Uint16**
   - Comment by @crotwell: Uint32 - with the caveat that datacenters may choose to limit input/output record sizes based on their needs.

Requirement: Provide nanosecond-timing resolution in both record start time and sample rate/period

https://github.com/FDSN/miniSEED3-TechnicalEvaluation/issues/16

Summary

compatibility of the number of samples field is implied

Questions

1. Do we want nano-second precision for the time-stamp? (Yes/No)
   - **9 x YES** | **0 X NO**
2. Do we want nano-second precision for the sampling rate? (Yes/No)
   - **9 x YES** | **0 X NO**
3. Should the definition state that the sampling rate is the nominal sampling rate? (Yes/No)
   - **0 x YES** | **9 X NO**
4. Should there be a flag to state the sampling rate is the nominal/calculated sampling rate? (Yes/No)
   - **0 x YES** | **9 X NO**

Requirement: Provide support for high sampling rates

https://github.com/FDSN/miniSEED3-TechnicalEvaluation/issues/17

Questions

Should the new format provide support for high-precision (double precision/64 bit) sampling rates? (Yes/No)

- **8 x YES** | **0 X NO** (1 x no objection to a yes)
- Comment by @ketchum-usgs: I do not know that a sample rate can be known to more than 1 part in about 10^7, so a double seems excessive. I do not object to increasing the precision, but it seems like a waste of 4 bytes.

Requirement: Move key, selected blockette details into fixed header

https://github.com/FDSN/miniSEED3-TechnicalEvaluation/issues/18

Question

Should the above mentioned key fields be moved into the fixed header? (Yes/No)

- **9 x YES** | **0 X NO**

Requirement: Fix byte order of binary portions of header; define byte order of payload via encoding values

https://github.com/FDSN/miniSEED3-TechnicalEvaluation/issues/19

Questions

1. Should the byte-order of the header be fixed? (Yes/No)
   - **9 x YES | 0 X NO**
2. Assuming yes on (1): What should it be? (Little endian / Big endian)
   - **5 x Little | 1 X Big with no large preference | 3 X no strong opinion**
3. Should the byte order of the data be defined by the encoding? Already implicit for some in miniSEED 2.x; others require some additional specification. (Yes/No)
   - **9 x YES | 0 X NO**

Requirement: Remove blockette support

https://github.com/FDSN/miniSEED3-TechnicalEvaluation/issues/20

Summary

This goes together with #14 and #7.

Questions

Should we remove the abstraction of blockettes and replace it with something in #14? (Yes/No)

- **9 x YES | 0 X NO**

Requirment: Simplify and improve record start time

https://github.com/FDSN/miniSEED3-TechnicalEvaluation/issues/21

Questions

Should the start time in each record always be the actual start time? E.g. any kind of time correction has already been applied. (Yes / No)

- **9 x YES | 0 X NO**

Requirement: Eliminate time correction field as a required, always-present field and retain as an optional field present when needed

https://github.com/FDSN/miniSEED3-TechnicalEvaluation/issues/22

Questions

1. (Assuming #21 gets a no) Should the time-correction field be optional? (Yes/No)
   - **7 x YES | 2 X NO**
   - Comment by @chad-iris: Yes. Assuming we have optional extra headers.
   - Comment by @kaestli: No (Assuming we avoid optional extra headers :)
   - Comment by @ketchum-usgs: yes, use optional header if time is not right
2. (Assuming #21 gets a yes) Should there be an optional flag "time-correction applied"?
   - Comment by @crowell: this is a properly a provenance issue, but a datacenter could increment #13 data/publication version if/when time corrections are applied.
   - **0 x YES | 9 X NO**

Requirement: Combine and drop unused / underspecified bit flags

https://github.com/FDSN/miniSEED3-TechnicalEvaluation/issues/23

Questions

Should unused/underspecified bit-flags be dropped? (Yes/No)

- **9 x YES | 0 X NO**
- Comment by @chad-iris:

```
I/O flags, Bit 0: station volume parity error possibly present
I/O flags, Bit 1: long record read
I/O flags, Bit 2: short record read
I/O flags, Bit 3: start of time series
I/O flags, Bit 4: end of time series



But to be honest all of them can be retained if we have optional header capability. With some marked as deprecated if we wish to discourage their use. The ones above are just the
```

Requirment: Eliminate sequence number as a required, always-present field

https://github.com/FDSN/miniSEED3-TechnicalEvaluation/issues/24

Questions

Should the sequence number be demoted to an optional field only present when needed? (Yes/No)

- **9 x YES | 0 X NO**
- Comment by @kaestli: they should actually be removed at all

Requirement: Alongside with the new format, a protocol for (near) real time data exchange needs to be specified

https://github.com/FDSN/miniSEED3-TechnicalEvaluation/issues/25

Questions

1. Should suitability for real-time transfer be a major design goal of the new format? Otherwise it would be designed to be as suitable for real-time transfer as reasonable without compromising other use cases. (Yes/No)
    - **2 x YES | 7 X NO**
2. Assuming a no on (1): Should some fixed header fields be moved to a footer? (Yes/No)
    - **1 x YES | 8 X NO**
3. Should a different approach like the mentioned "sub record streaming" be investigated in more detail? (Yes/No)
    - Consensus is that it should be evaluated but not in the scope of the NGF.