WPEC EG-GNDS Status

28 June 2019

OECD-NEA, Boulogne-Billencourt, France

Sep. 2016 Science and Technology Review, LLNL, https://str.llnl.gov/september-2016/beck

Several meetings since WPEC 2018

- (US Only) unofficial SG-43/EG-GNDS meeting, ORNL, Feb. 2019
- ND2019, China National Convention Center, Beijing, China, 23 May 2019
- NEA Headquarters, Boulogne-Billancourt, France, 25 June 2019

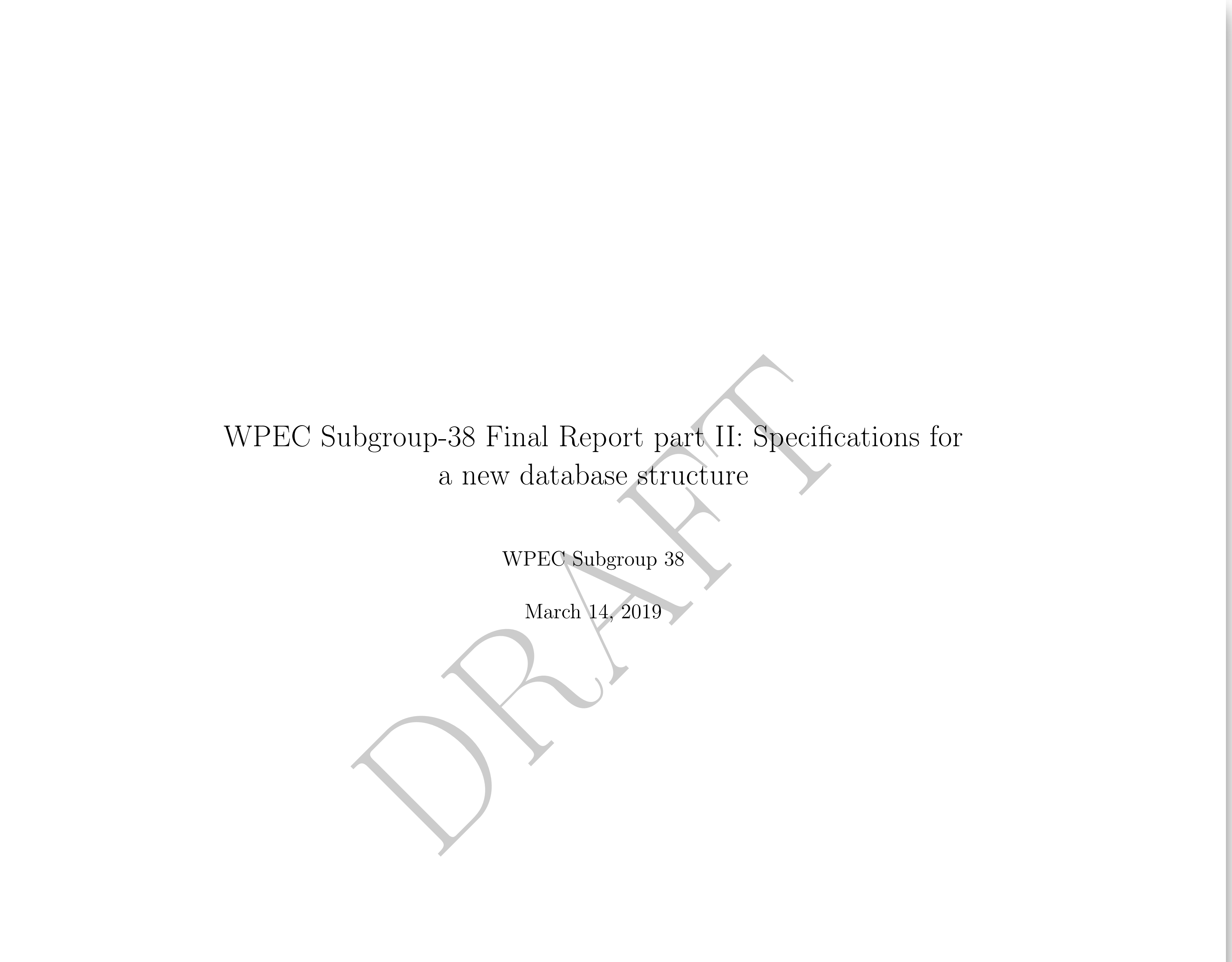
AGENDA

Start	End	Topic	Participant(s)	Country
14:00	14:10	Welcome and introductions	David BROWN	UNITED STATES
14:10	14:20	Review of mailing list and membership	Michael FLEMING	NEA
14:20	14:40	Review of summary record and actions	David BROWN, Michael FLEMING	UNITED STATES, NEA
14:45	15:00	Status of the specifications for GNDS-1.9	David BROWN	UNITED STATES
15:00	15:30	Discussion and approval of version 1.9	All	
15:30	15:45	**Coffee Break**		
15:45	16:00	NEA GitLab and GNDS format proposal process	Michael FLEMING	NEA
16:00	16:30	Discussion on proposal/approval processes, GitLab Q&A	All	
16:30	16:50	Report on SG43 activities	Jeremy CONLIN, Caleb MATTOON	UNITED STATES
16:50	17:10	Review of GNDS support in codes	All	
17:10	17:30	Outreach and tutorials	TBD	
17:30	17:50	Nuclear data structure and format: applications or science driven?	Jean-Christophe SUBLET	IAEA
17:50	18:00	Any other business	All	

At WPEC EG-GNDS Meeting NEA, OECD Conference Centre, 15 May 2018, we agreed to several things

- GNDS-1.9 is first official version EG-GNDS will “bless”
- GNDS-1.9 needs complete specifications
- We will maintain GNDS using NEA’s Gitlab
- We will develop format improvement mechanism modeled on the operations of CSEWG’s ENDF Formats Committee

Draft specifications for GNDS-1.9 are ready



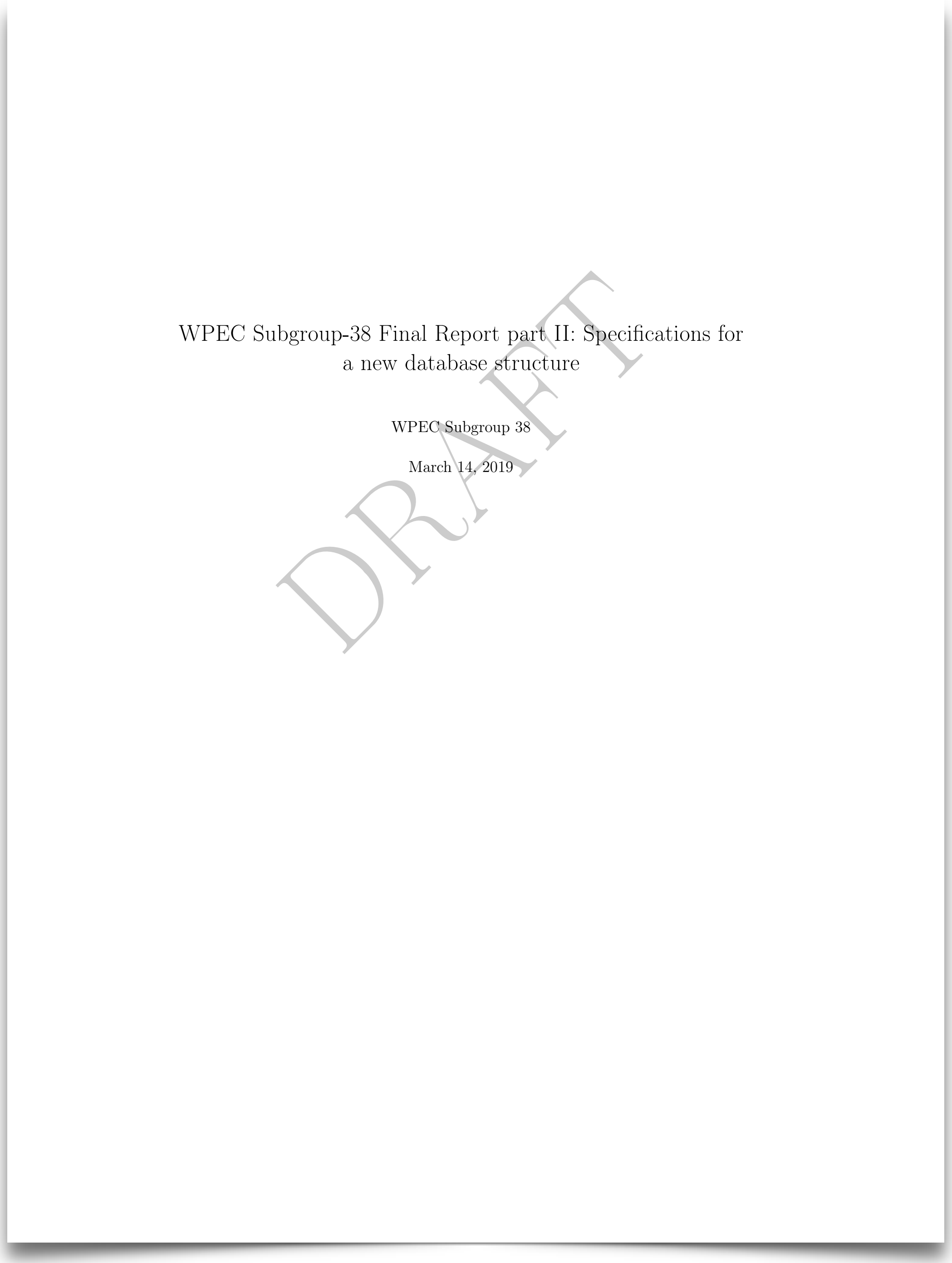
WPEC Subgroup-38 Final Report part II: Specifications for a new database structure

WPEC Subgroup 38

March 14, 2019

Specifications doc. also cover series of “format proposals”

- **Several GNDS-1.9 formats are “quick-n-dirty” translation of ENDF-6, missing requested features:**
  - TSL
  - FPY
- **There are many loose ends**
  - Resonances
  - Covariances
- **Requirements mention several things we have not implemented**
  - Documentation
  - Several processed data forms



WPEC Subgroup-38 Final Report part II: Specifications for a new database structure

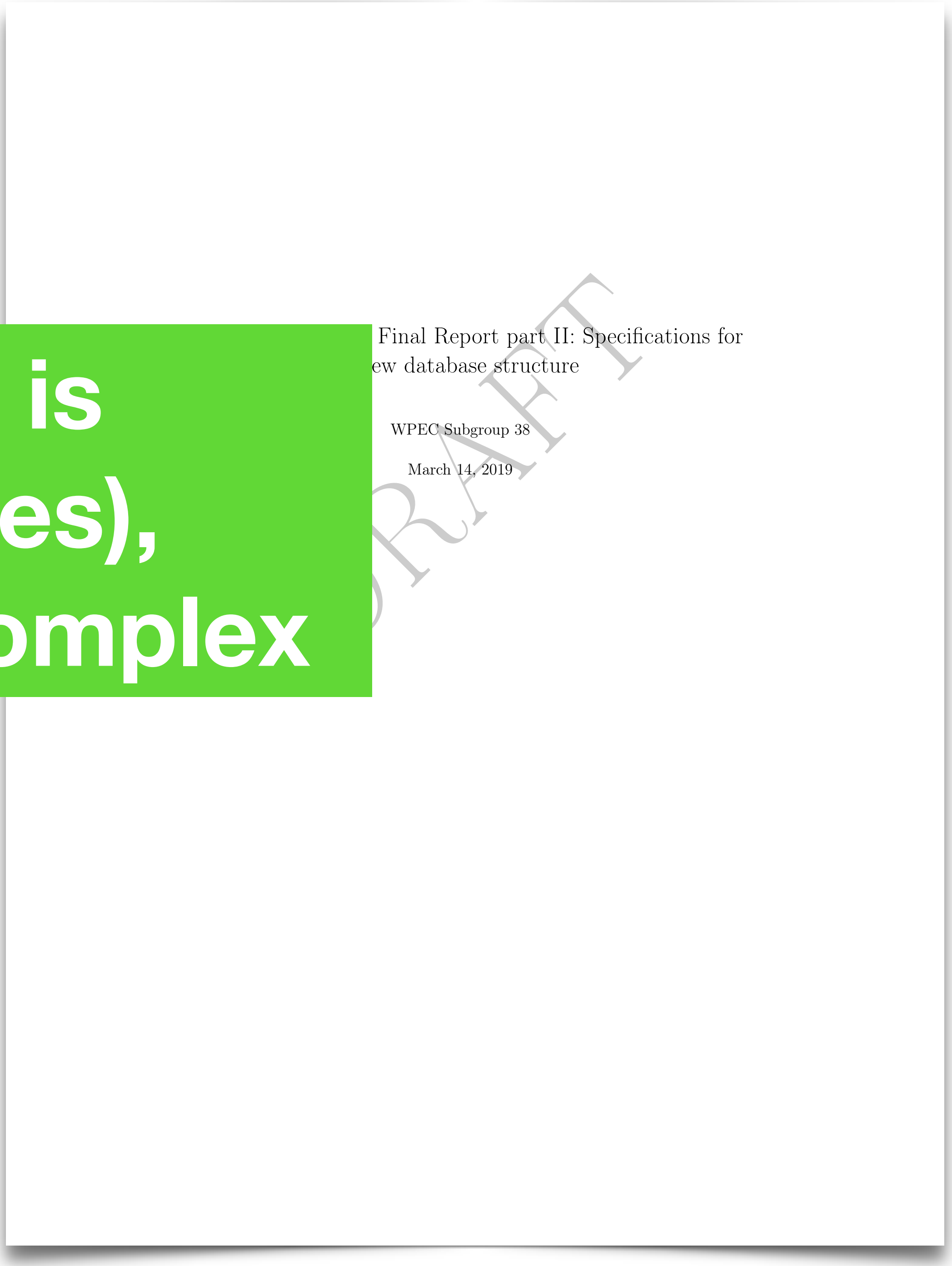
WPEC Subgroup 38

March 14, 2019

Specifications doc. also cover series of “format proposals”

- **Several GNDS-1.9 formats are “quick-n-dirty” translation of ENDF-6, missing requested features:**
    - TSL
    - FPY
- **There are many**
    - Resonances
    - Covariances
- **Requirements mention several things we have not implemented**
    - Documentation
    - Several processed data forms

The format is big (385 pages), beautiful and complex



Final Report part II: Specifications for
ew database structure

WPEC Subgroup 38

March 14, 2019

What next?

- We can’t approve the GNDS-1.9 specifications as is, can we?
- Very few of us have read it all, how can we make an informed decision?
- But, the document will always have typos, things to tweak, etc., so in a sense *it will never be done*

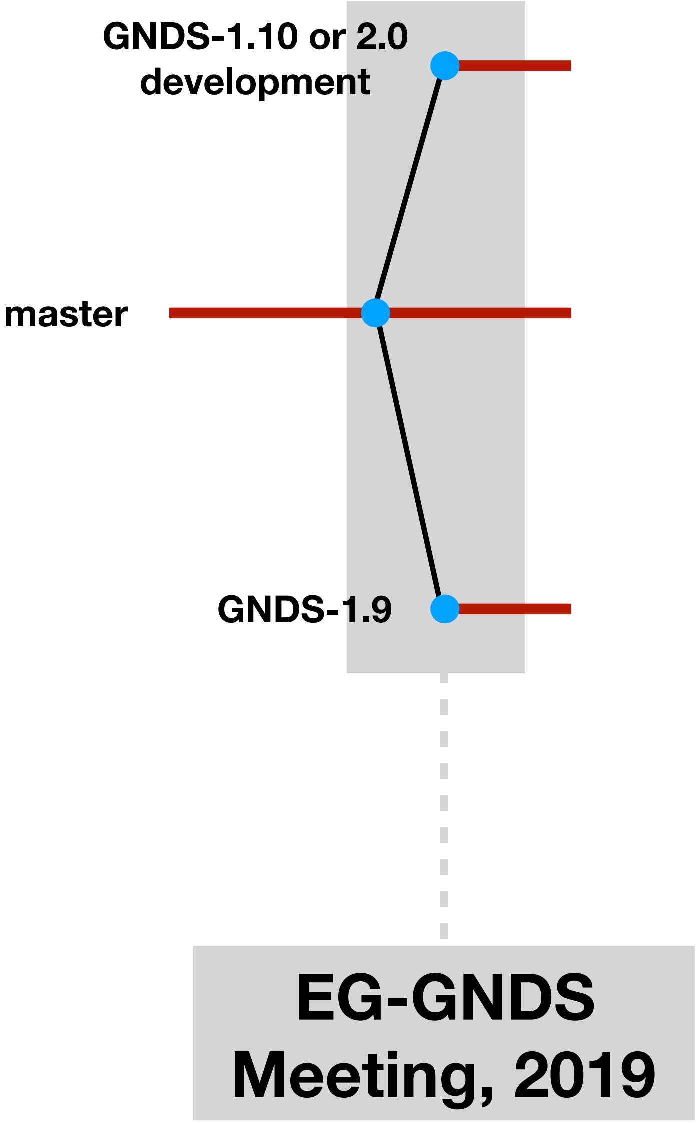
- Only LLNL has implemented the whole thing and LLNL is finding errors in their implementation *and* they already want to move beyond GNDS-1.9

Suggestion:

- **approve 1.9 anyway, *and***
- **engineer a mechanism for continuous improvement**

Discussion and approval of version 1.9 and future revisions

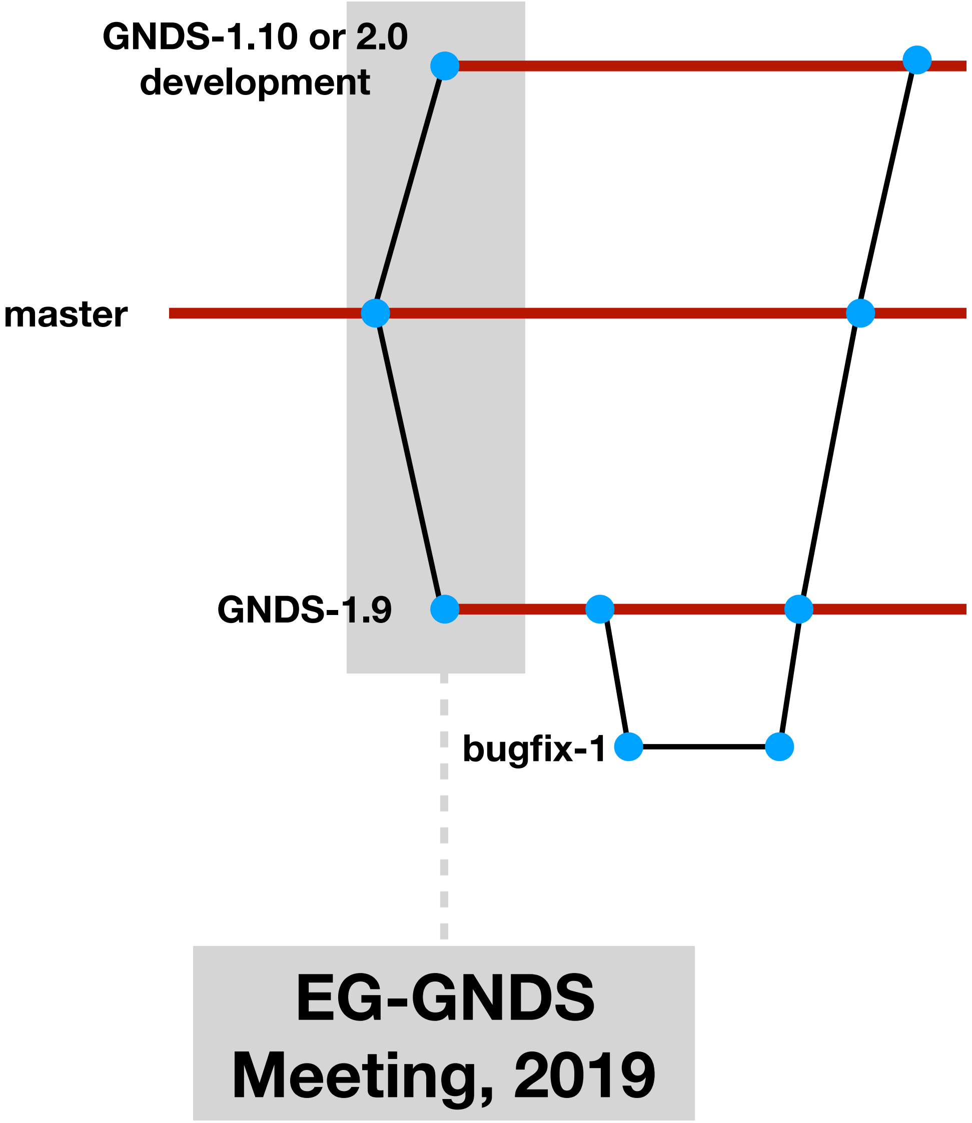
Harness the power of git



I propose we

- approve GNDS-1.9, *as is*, today
- make 2 git branches
- *lock them*

Harness the power of git



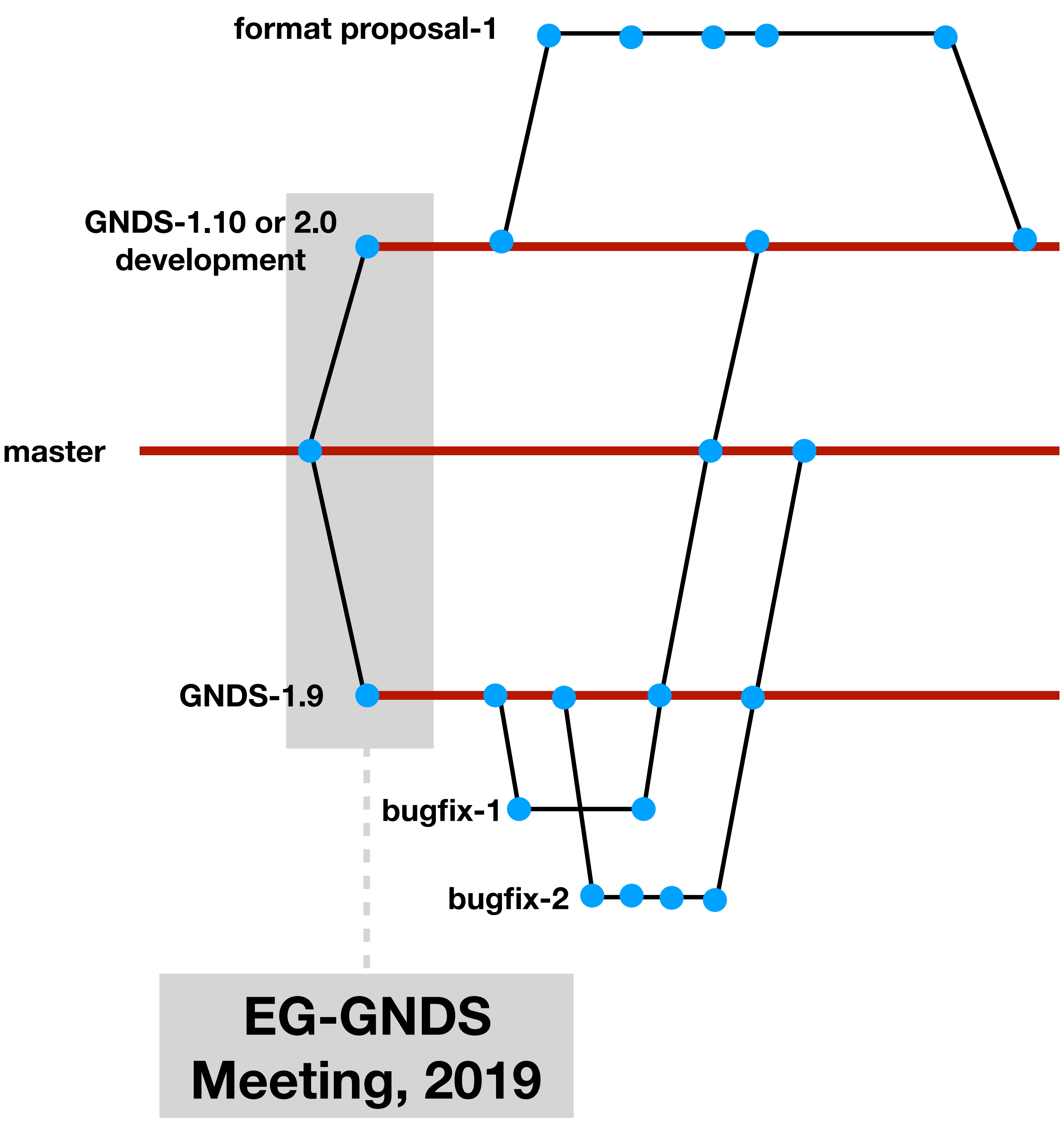
I propose we

- approve GNDS-1.9, *as is*, today
- make 2 git branches
- *lock them*

Furthermore,

- use git's branch and merge mechanism to manage fixes

Harness the power of git



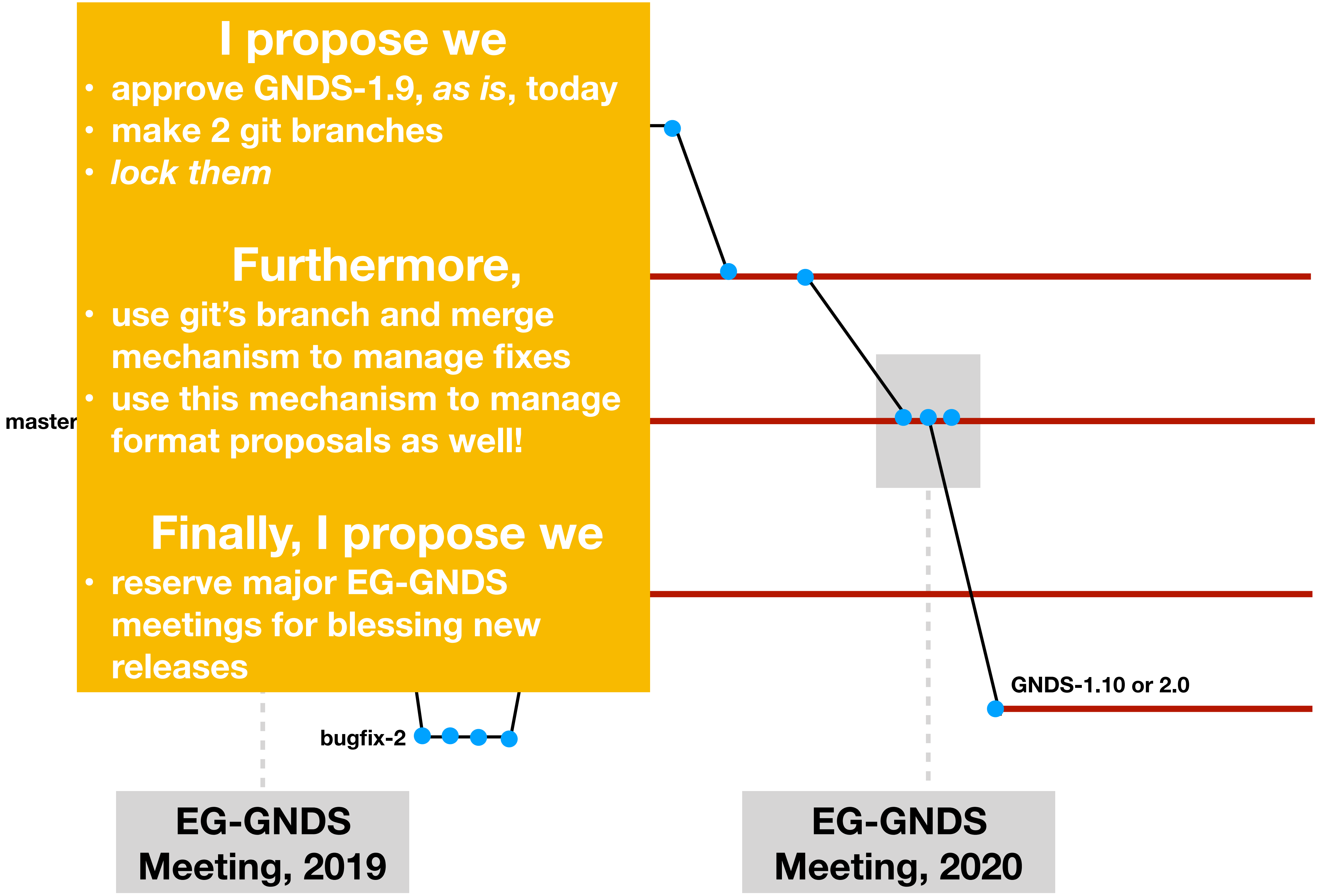
I propose we

- approve GNDS-1.9, *as is*, today
- make 2 git branches
- *lock them*

Furthermore,

- use git's branch and merge mechanism to manage fixes
- use this mechanism to manage format proposals as well!

Harness the power of git



I propose we

- approve GNDS-1.9, *as is*, today
- make 2 git branches
- *lock them*

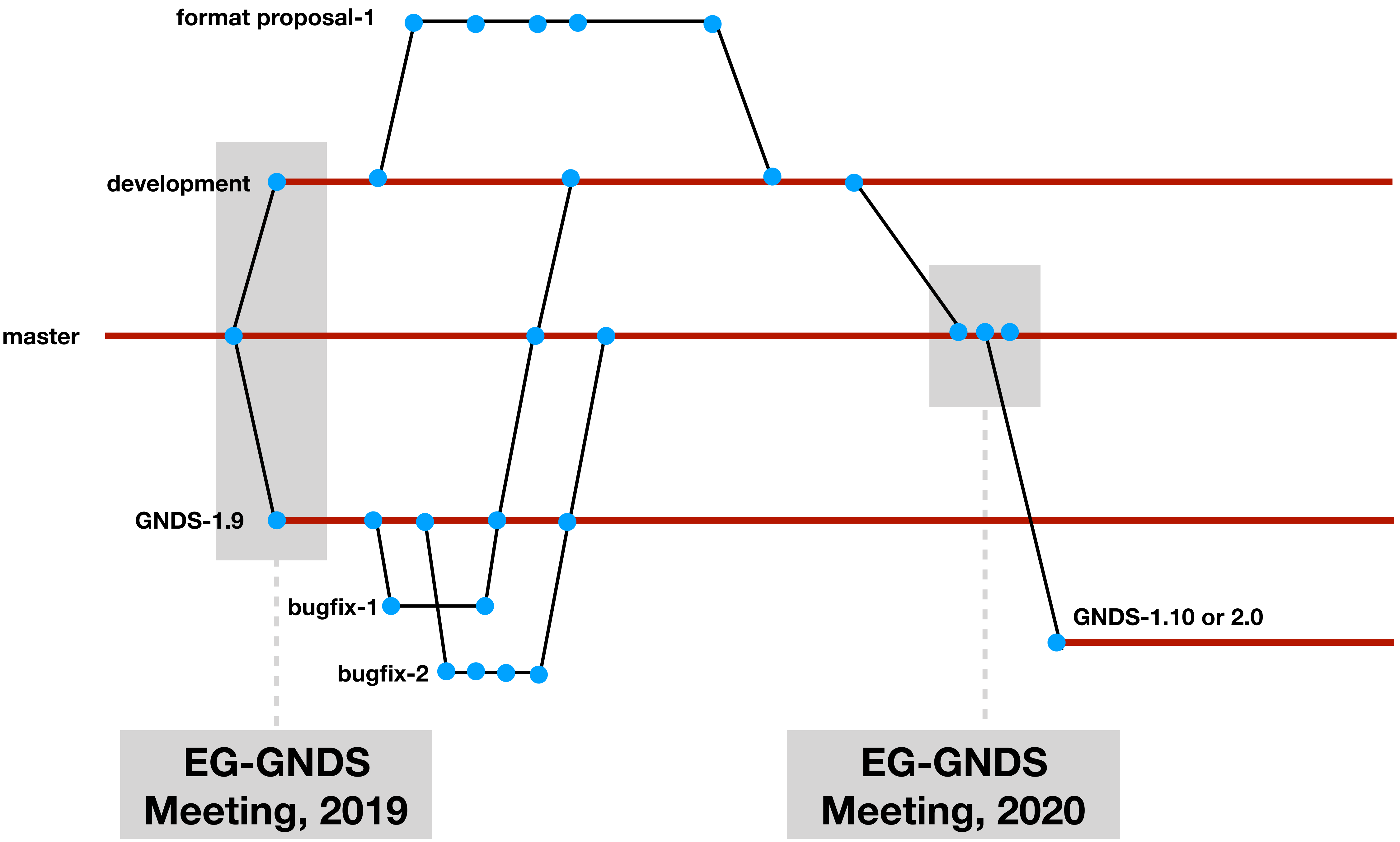
Furthermore,

- use git’s branch and merge mechanism to manage fixes
- use this mechanism to manage format proposals as well!

Finally, I propose we

- reserve major EG-GNDS meetings for blessing new releases

Harness the power of git



In summary

I propose we

- approve GNDS-1.9, *as is*
- make 2 git branches
- *lock them*

Furthermore,

- use git's branch and merge mechanism to manage fixes
- use this mechanism to manage format proposals as well!

Finally, I propose we

- reserve major EG-GNDS meetings for blessing new releases

Consensus was: This is a good development model and GNDS-1.9 is imperfect, but we have to start somewhere!

What constitutes a format proposal vs. a “bugfix”?

- Typos (OK),
- Wrong equations (maybe),
- New formats (No)

It is very dependent on the nature of the change

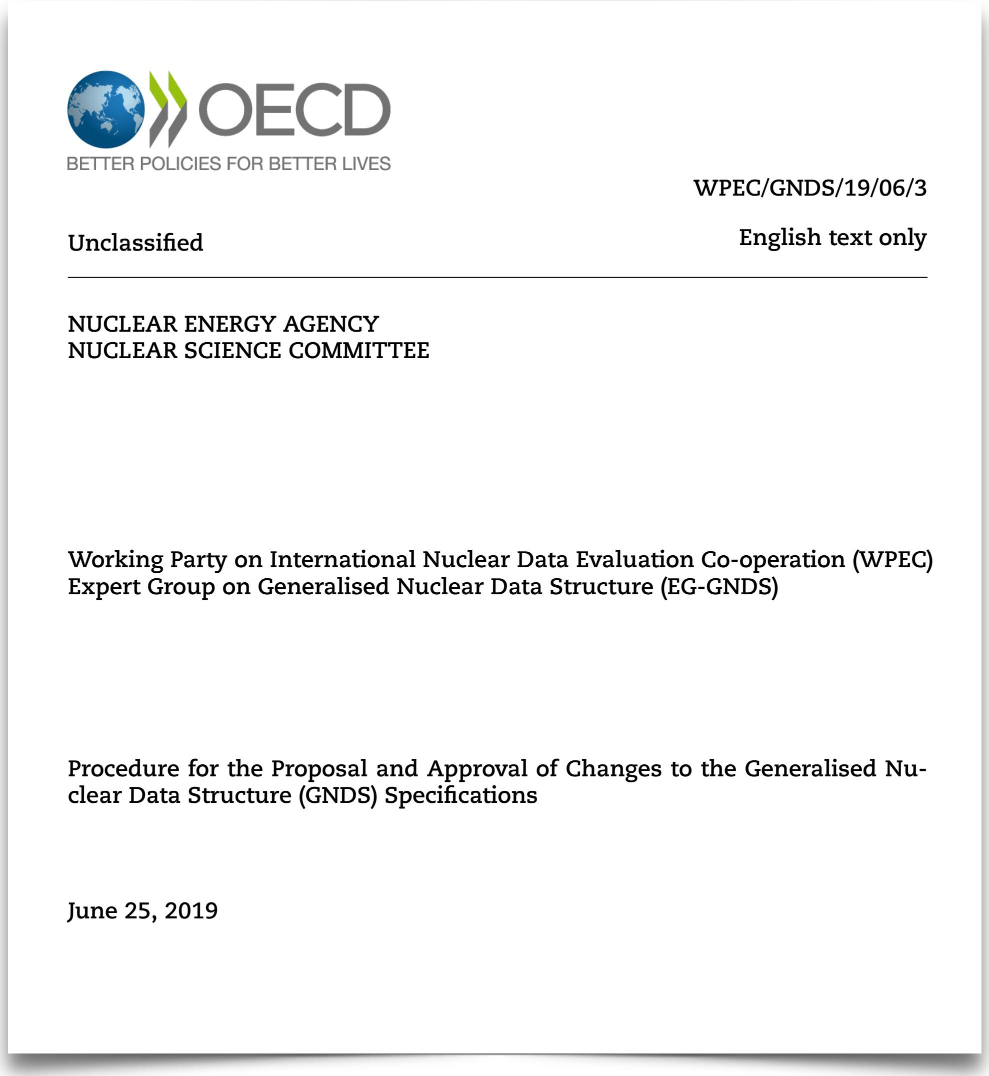
Consensus view: It’s the chair’s job to sort it out. If we get it wrong then we’ll invest in a formal procedure.

Creating a format proposal

- **Bare minimum format proposal for next release (GNDS-1.10 or 2.0):**
  - Latex descriptive text
  - JSON changes (these actually encode the format)
  - Also, has to correctly build to a PDF.
- **In future releases we will discuss other requirements such as:**
  - Reference implementation in a code
  - An example file
  - Code /stylesheet to translate back and forth

Approving a format proposal

- Format proposal get their own branch and must pass the CI. Submission of proposal done with “merge request” on gitlab
- Chair & EG nominate reviewers (self nomination allowed!)
- One or more reviewers interact with proposers through gitlab discussion board to resolve issues
- Chair performs the merge



OECD
BETTER POLICIES FOR BETTER LIVES

WPEC/GNDS/19/06/3

Unclassified

English text only

NUCLEAR ENERGY AGENCY
NUCLEAR SCIENCE COMMITTEE

Working Party on International Nuclear Data Evaluation Co-operation (WPEC)
Expert Group on Generalised Nuclear Data Structure (EG-GNDS)

Procedure for the Proposal and Approval of Changes to the Generalised Nuclear Data Structure (GNDS) Specifications

June 25, 2019

Thank you Michael!

Problems with this mechanism

- **The structure of the format is contained in a JSON-formatted meta-format (it isn't scary)**
- **The PDF is built from frame LaTeX documents & automatically generated text from meta-format (not scary in principal)**
- **In the rush to put the whole system together, the project is a complex, fragile mess**
  - *ACTION: C. Mattoon & D. Brown to clean up project*

Until project is cleaned up & organized better, changes should be done in collaboration with someone who knows how system works: D. Brown, C. Mattoon, J. Conlin, W. Haeck, D. Wiarda, M. Fleming…

Other questions

- **How would an eventual NEA document versioning align with this model?**
    - ACTION: *M. Fleming to investigate*
- **What reviews are needed for final documents at each or our institutions?**
    - ACTION: *All EG members need to investigate, but in US there is clear process*
- **With revision model, won't there be a proliferation of formats?**
    - Yes! At least one/year. This is what happens in ENDF, no one cared before because format only declared final once ENDF library is released.
    - Want to see the process in action to see if this is a problem
- **What to do in the event of a conflict?**
    - Surprisingly this hasn't happened yet! So far potential conflicts quickly turned into collaboration on Gitlab.
    - If conflict can't be resolved, then the annual EG-GNDS meeting is the place to resolve?

Housekeeping Actions

- ACTION: (D. Brown) Investigate “latexdiff” for use in comparing versions of specifications

- ACTION: (D. Brown) Clean up FIXME’s, TODO’s, DRAFT’s etc, turn off GNDS-1.10 items for clean GNDS-1.9 file

- ACTION: (C. Mattoon, D. Brown, M. Fleming) Prepare branch tree for future

- ACTION: (M. Fleming) Orchestrate formal email vote on GNDS-1.9 specifications once ready

- ACTION: (D. Brown) Investigate Latex->HTML build options

Outreach and tutorials

Pros

- We're getting asked for information
- We have specifications
- LLNL has C++ APIs: https://github.com/LLNL/gidiplus
- We have a full library (ENDF/B-VIII.0) in GNDS-1.9

Cons

- Do we have a champion/lead for this activity?
- Current LLNL Python APIs are not publicly available yet. Old version available at BNL: https://www.nndc.bnl.gov/endf/codes/FUDGE/index.html

** ACTION: Bret Beck volunteers to spearhead!*
Plan ANS or similar workshop