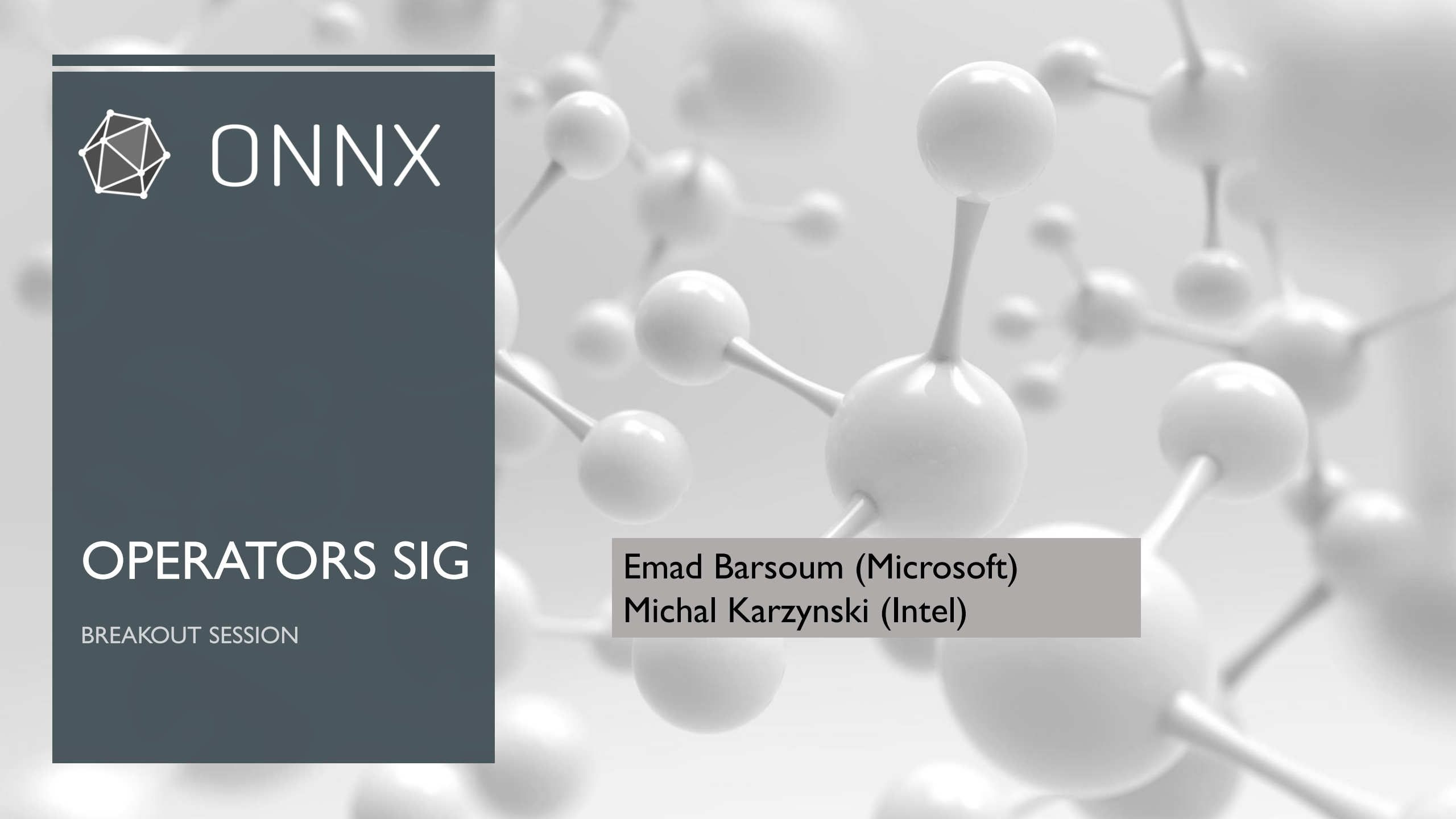


ONNX

OPERATORS SIG

BREAKOUT SESSION

Emad Barsoum (Microsoft)
Michal Karzynski (Intel)

AGENDA

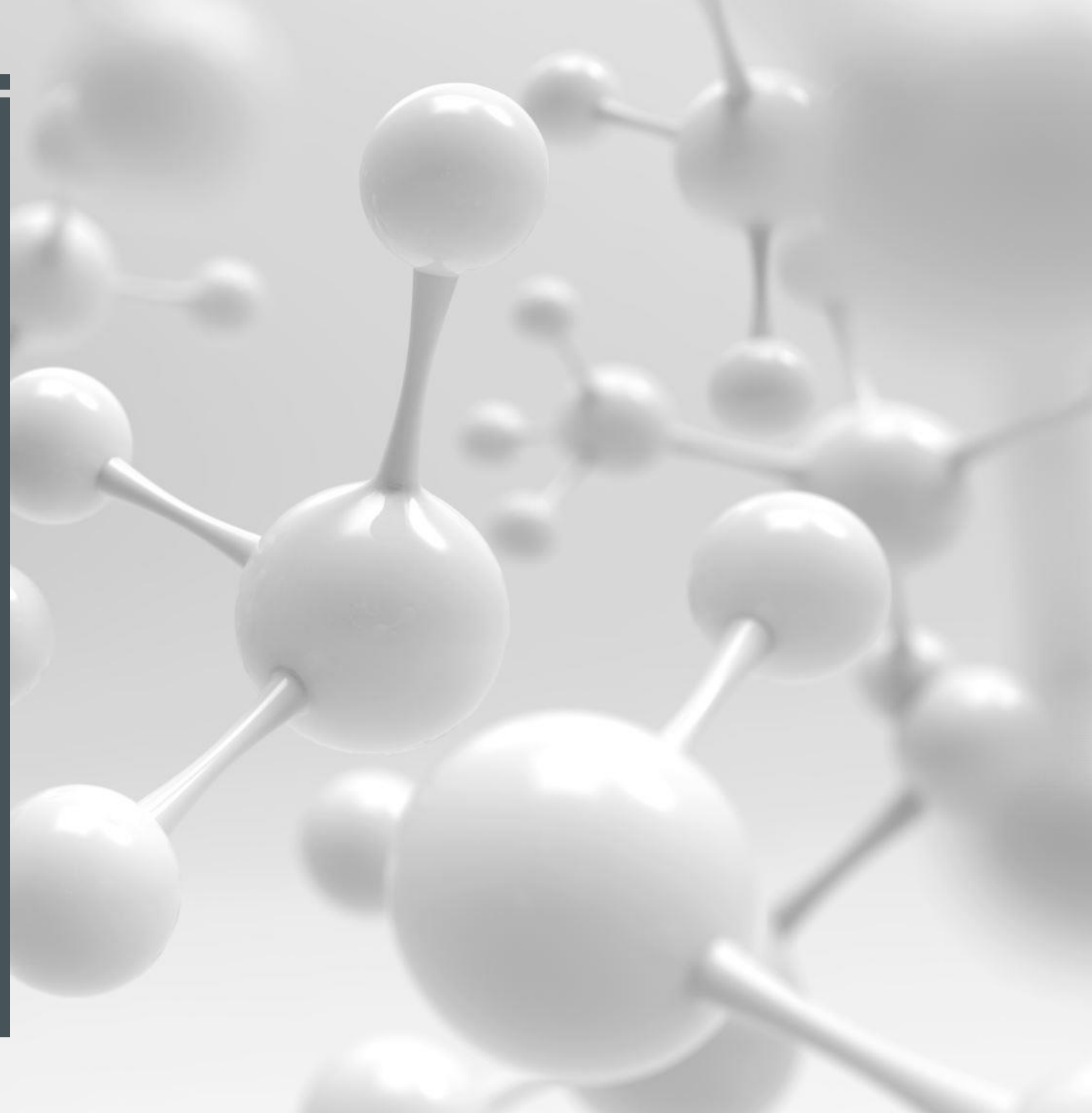
- Operators SIG
- Add new operator update
- Proposal / improvement
- Discussion: Version converter
- Discussion: PR and Issues

GOAL

Keep Up	Quality	Clarity	Size	PRs and Issues
Keep up with the latest progress in AI	Improve the quality of ONNX Operators	Reduce ambiguity	Avoid bloating ONNX spec	Keep up with PRs and operator issues

PARTICIPANTS

- Akinlawon Solomon (Qualcomm)
- Darren Crews (Intel)
- Dilip Sequeira (NVidia)
- Ganesan Ramalingam (Microsoft)
- Itay Hubara (Habana)
- Jianhao Zhang (JD)
- Ke Zhang (Alibaba)
- Leonid Goldgeisser (Habana)
- Milan Oljaca (Qualcomm)
- Ofer Rosenberg (Qualcomm)
- Rajeev K Nalawadi (Intel)
- Scott Cyphers (Intel)
- Shlomo Raikin (Habana)
- Simon Long (GraphCore)
- Spandan Tiwari (Microsoft)
- Wei-Sheng Chin (Microsoft)
- Weiming Zhao (Alibaba)
- Liqun Fu (Microsoft)



COMMUNICATION

- Slack channel: https://slack.lfai.foundation and join onnx-operators
- Discussions on GitHub PRs and issues
- Meetings announcement are on Slack and Gitter
- Docs and meeting notes are in onnx/sigs https://github.com/onnx/sigs/tree/master/operators
- Deprecated: Gitter channel: https://gitter.im/onnx/operators

ADDING NEW OPERATOR ISSUE

- Reference implementation in Python isn't enough.
- Runtime and framework writers, start implementing new operators close or after ONNX release.
  - Any issue cause delay to the release.
  - Or worse, cause a patch release.
- Operator behavior might not match existing framework, especially in corner cases.

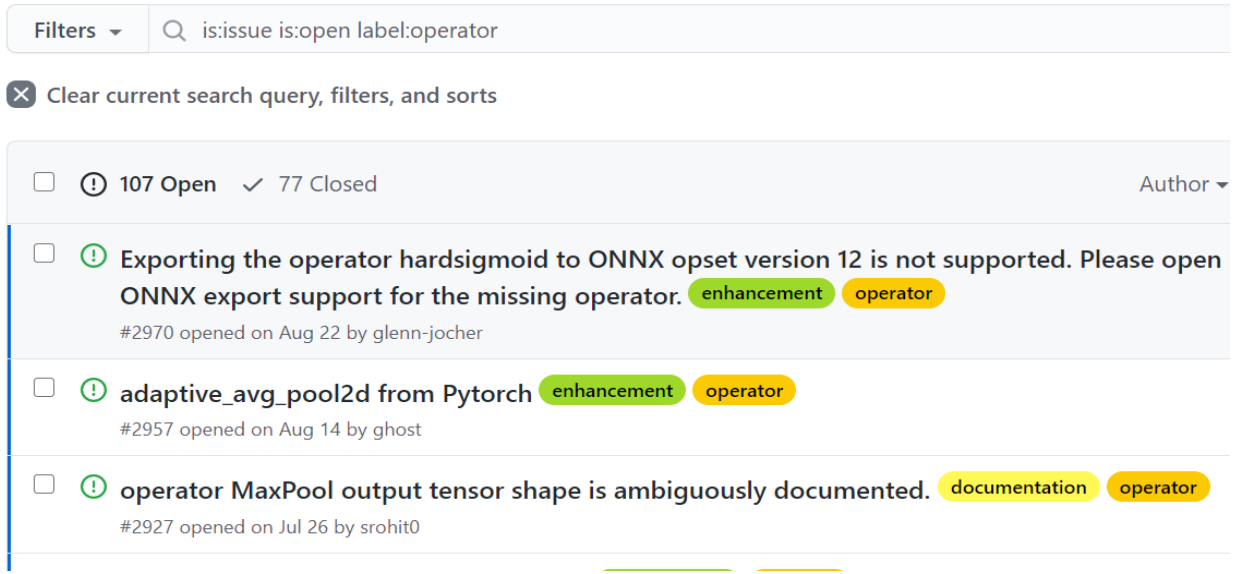
ADDING NEW OPERATOR UPDATE

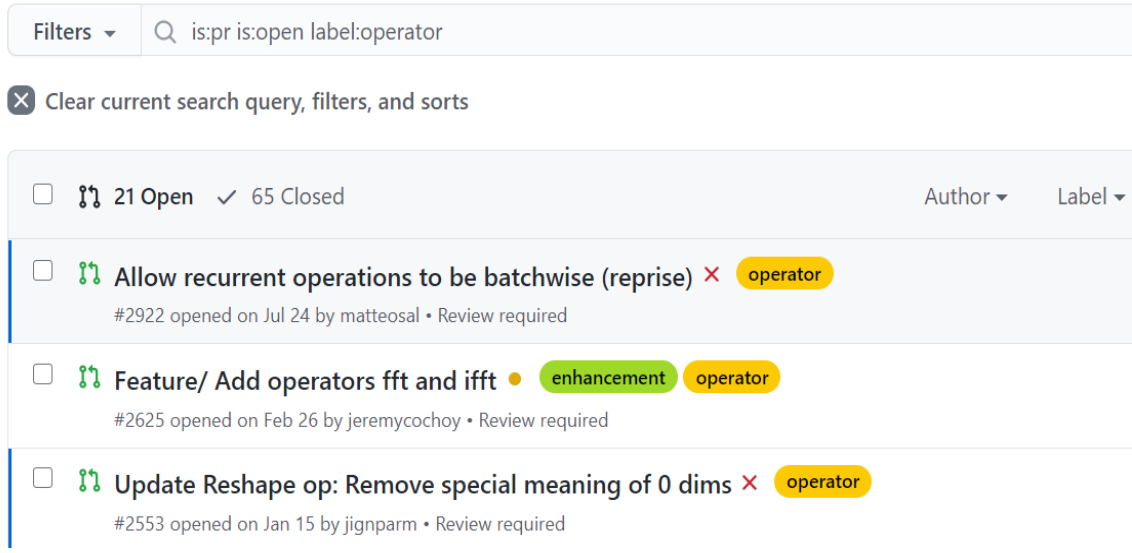
- Unit tests need to have the same coverage as the original framework.
- Test data need to be generated from the original framework to match behavior.
- [Optional] verify the new operator/function in a runtime/framework that support ONNX.

PROPOSAL

- Feel free to propose any improvements, such as:
  - Better testing, validation and coverage of ONNX operators.
  - Better documentation generation.
  - More operators.
  - A lot of existing manual steps need automation.
- For any big proposal, you will be invited in the SIG meeting to present it.

GET INVOLVED:
SUBMIT AND REVIEW PRS





PR REVIEW

- PRs should be marked with the Operator label
  - https://github.com/onnx/onnx/pulls?q=is:pr+is:open+label:operator
- Ops Contributors Group should review the PRs according to guidelines
- Mature PRs can be discussed during bi-weekly sync
- Final approval by member of SIG-operators-approvers group

GITHUB DISCUSSION ISSUES

- Many open discussions marked with Operator label
- Ops Contributors Group should be active in discussions and encourage submission of PRs
- We should decide which discussions can be closed
- Still looking for best way to triage this large number of open issues

VERSION CONVERTER

- Used to convert from one OPSet to another or vice versa.
- Currently, it is outdated.
- Should we enforce every operator PR to update the version converter?

TIME MAJOR FLAG FOR RECURRENT

- Issue and PR:
  - https://github.com/onnx/onnx/issues/2159
  - https://github.com/onnx/onnx/pull/2922
  - https://github.com/onnx/onnx/pull/2284
- Current recurrent operator in ONNX, operate on:
  - [seq_length, batch_size, input_size]
- Most framework support:
  - [seq_length, batch_size, input_size] and [batch_size, seq_length, input_size]
  - Some framework hide the batch axis.

THANKS FOR COMING!!!

Operator SIG resources

- Slack channel: https://slack.lfai.foundation and join onnx-operators
- Documents and artifacts: https://github.com/onnx/sigs/tree/master/operators