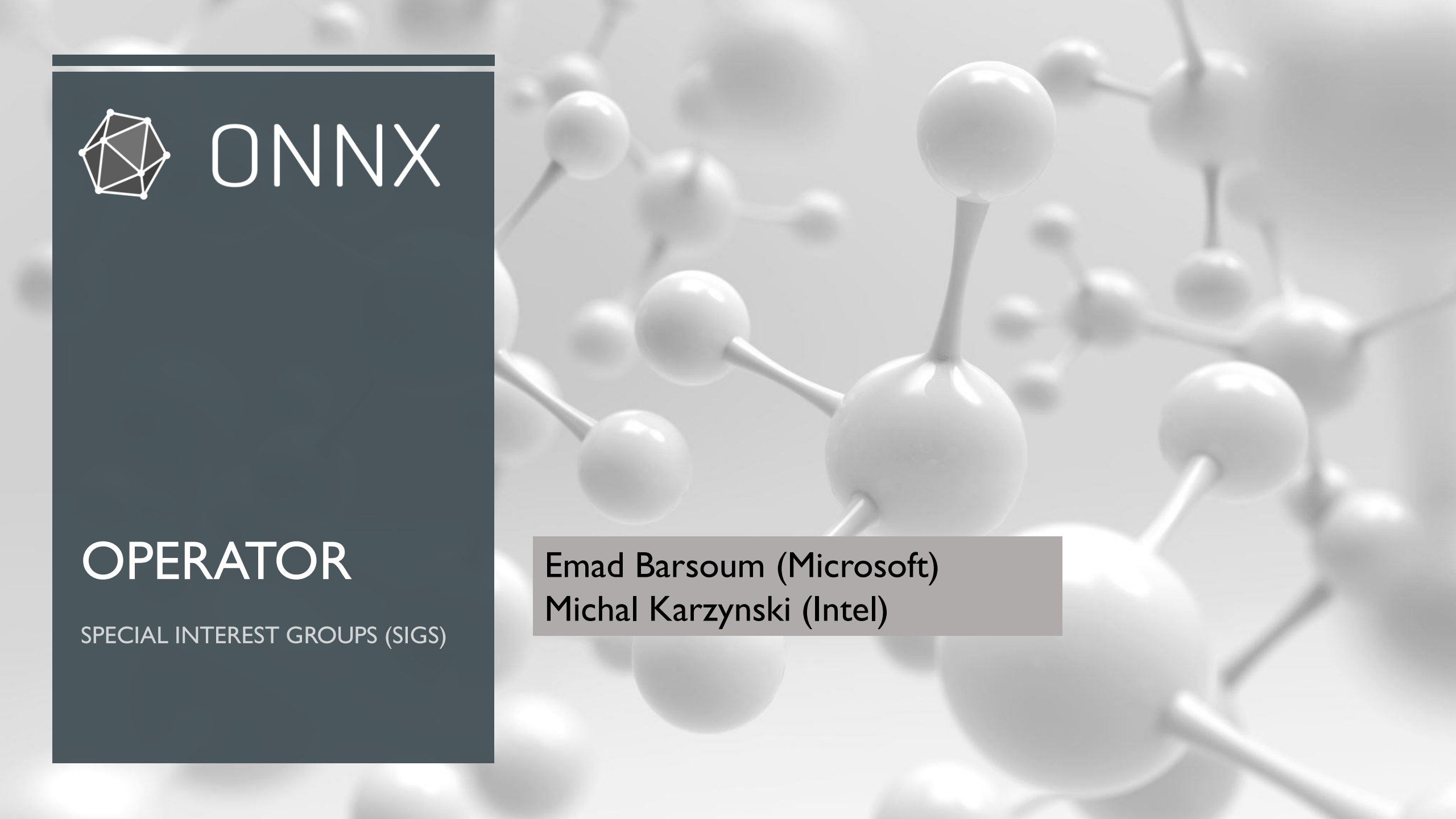


ONNX

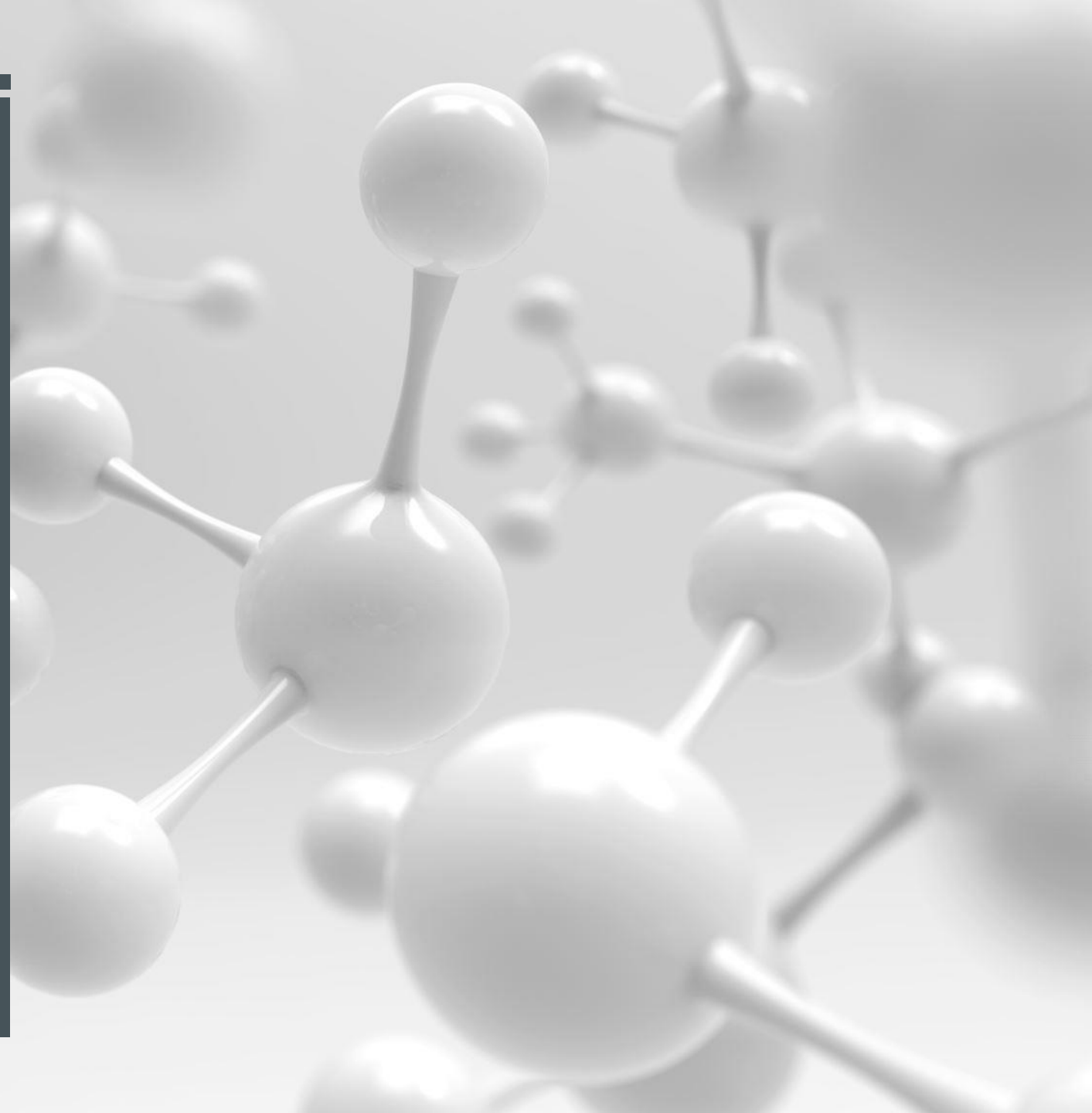
OPERATOR

SPECIAL INTEREST GROUPS (SIGS)

Emad Barsoum (Microsoft)
Michal Karzynski (Intel)

PARTICIPANTS

- Darren S Crews (Intel)
- Dilip Sequeira (NVidia)
- Ganesan Ramalingam (Microsoft)
- Jianhao Zhang (JD)
- Prasanth Pulavarthi (Microsoft)
- Wei-Sheng Chin (Microsoft)
- Ke Zhang (Microsoft)
- Milan Oljaca (Qualcomm)
- Spandan Tiwari (Microsoft)
- Rajeev K Nalawadi (Intel)
- Simon Long (GraphCore)



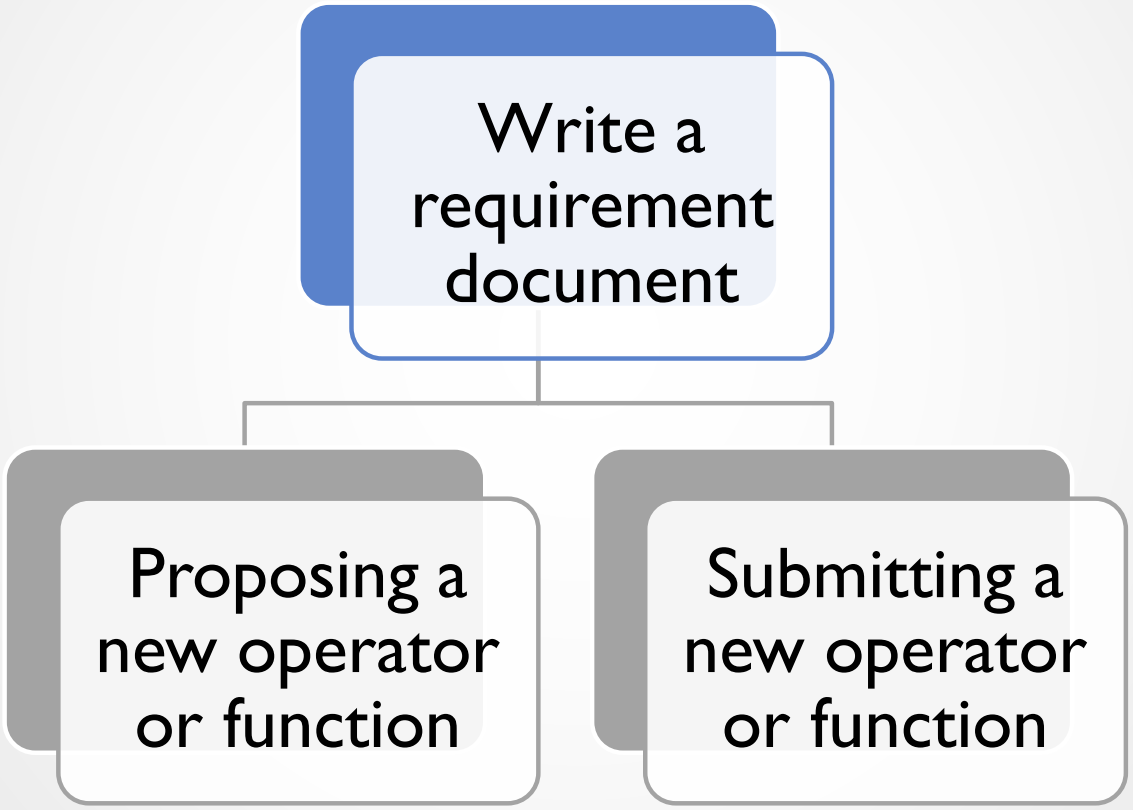
LOGISTICS

- Start with a biweekly meeting.
- Use gitter operator channel for all our online discussion
  - https://gitter.im/onnx/operators
- Use online world for discussion and proposal draft.
- Use markup for final document.
- All our drafts, meeting links and proposals are shared in gitter.

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

FIRST TASK


Write a requirement document
Proposing a new operator or function
Submitting a new operator or function

PROPOSING A NEW OPERATOR

- If the operator can be composed by other ONNX operators, then it should be a function and not an operator.
- If the operators can be split to new primitives, propose those primitives instead and make the operator a function.
- Based on a model.
- The operator needs to be implemented by at-least one (well-known) framework.
- Operator signature and behavior:
  - If the operator is available in numpy, prefer numpy semantics.
  - If the operator is available in more than one frameworks, make sure that your design is general and cover those frameworks.
- Prefer attributes over inputs.

SUBMITTING A NEW OPERATOR OR FUNCTION

- Description:
  - Write a detailed description about the operator, and its expected behavior. Pretty much, the description should be clear enough to avoid confusion between implementors.
  - Add an example in the description to illustrate the usage.
  - Add reference to the source of the operator in the corresponding framework in the description (if possible).
  - Write the mathematic formula or a pseudocode in the description. The core algorithm needs to be very clear.
- Write a reference implementation in Python, this reference implementation should cover all the expected behavior of the operator.
- Write unit test, that cover main usage and corner cases.
- Update the documentation.
- At least two sign-off from the operator contributor group.

PROPOSING AND SUBMITTING NEW OP

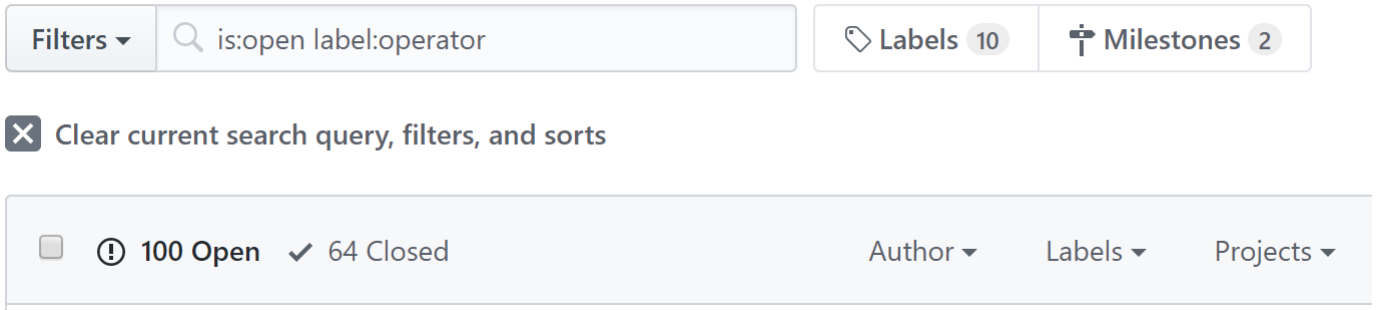
- Final document is already published to:
  - https://github.com/onnx/onnx/blob/master/docs/AddNewOp.md
- All reviewers will use the above document as submission criteria

OPERATOR CONTRIBUTOR GROUP

- A list of people who previously contributed to ONNX.
- Any operator PR need at least two signoff from this group.
- Based on people contributions, we will add or remove people from the group.

CURRENT FOCUS

- We have a lot of operator PRs and issues that need to be addressed.
- We will start reviewing PRs and issues in our regular meeting.
- We will start to assign them to reviewers.



THANKS FOR COMING!!!

Operator SIG resources

- Gitter channel https://gitter.im/onnx/operators
- Documents and artifacts: https://github.com/onnx/sigs/tree/master/operators