The TinyML computing stack is complex! As we just discussed there are:

- Lots of different model optimization methods
- Lots of specialized compilers for embedded devices
- Lots of inference frameworks for embedded deployment
- Lots of software kernels for unlocking the hardware’s performance
- Lots of specialized hardware and ML accelerators

Which of these causes of complexity do you think will have the greatest impact on the kinds of TinyML applications you are excited to deploy at scale? Which will not be an issue? How might you begin to approach solving some of this complexity?