

- Background mailing list
  - <https://groups.google.com/d/msg/kubernetes-wg-contribex/Aft7pTGSEUg/SbMqMfrSBQAJ>
- Topics
  - Organize code (maybe break pieces out)
  - Extension points needed
    - Some things 10% done but not usable
  - Define what belongs in core
- PR graph for commits per week
  - Jan 150 per week
  - Today around the same
    - 1k people
- Only way to scale is to split
  - Homebrew has x2 from k8tes
  - No others repos are matching the “speed of light” on GH
- Dashboards
  - Review and merge-queue latency
    - Find here <http://velodrome.kubernetes.io/>
    - Shows 2 weeks per commit
- 100 more issues per week, not scalable since April
- 80 PRs in flight per week prior to 1.0
  - 100 more with every release to backlog
- Github is designed for dedicated team for one project chunked repos
  - Fighting with creating more tooling to route
  - Can't add a ton of individuals to give full reign
- Why to split?
  - Clean separation of components
  - Can maybe help with issues
    - Less things for maintainer to subscribe to
  - Doc fixes don't need to run e2e
- Anecdotes from other repos
  - Cops has 3 maintainers and goal is 10 PRs per day
  - No other opinions
- Obvious problems
  - How do we bring a union build together with more repos
  - Can we make sure the experience is still friendly across others
  - Code-gen needs to have a handle
- Go client is underway for new repo
  - Should be in 1.5
- How can we migrate history + issues to the new repo
  - Found a solution
  - Test infra needs moved
  - Milestones need moved

- Bot work allows us to move between repos
  - Need more skeleton repo creation for common components
  - Need for meta-bot to watch for new repos and correct inconsistencies
- Concrete steps so far
  - Client library
    - No need to vendor everything
- Low hanging
  - Flaky tests
  - Docs
    - Should be priority
  - Example directories
  - Cloud provider (maybe)
  - Someone needs to spend analysis on churn points
  - Kubeadm
  - Minikube
- How can a non-Gogler help
  - Need to have process to create new repo
- How does release process work
  - All need to come together
  - Require owner
  - How do we version in this new world?
    - How do I mix/match versions
      - +/- 1 version needs to work
  - Vendoring
    - Go code has solution
    - Non go code needs owner to figure out
- Projected org/repos
  - Core
    - Kubectrl
    - Kubelet
    - Scheduler

- Extension mechanisms
  - API server federation
    - How can I expose new apis
    - How can I run on top instead of linking
    - How can I not break federation too
  - Authorization

- Admission control hooks
- Initializers + finalizers
  - Allow for deferred initializers
  - For more background watch “life of request talk”
- 3rd party resources
  - Lot of shared pain for feature gaps and bugs
- Kubectl extensions
  - Custom login flow dependencies
  - Add new subcommands easily
- Service brokers + operators
- Controller pattern
  - A lot have started to move into client library
- External cloudproviders
- CRI, network, storage plugins
- Cluster addons: UI, monitoring, logging
- Need help to link all issues to “next steps” for each components

#### Follow-up

- Probably not best for wg-contribex
- Probably good for new contribs to get involved to understand codebase better
- Special community hangout?
- API-machinery SIG
- Should SIGs start discussing themselves for docs?

#### What is Core?

- Concerns
  - Release bundle
  - What is shipped by default
  - CLuster addons Ex. kube-up brings dashboard up by default
  - Do we need conformance tests
  - Storage systems available
  - Kubectl extensions
  - Where is the proper place for extensions
- What is kubernetes for
  - Platform

- Need people to get involved to figure out what needs to split out based on use-cases
  - Need help from similar communities to figure out good/bad for similar projects
- Proposal for maintainer policy
  - Positive things about culture that needs to be preserved
  - New contribs need to be known
- Ideas
  - Core vs crust
  - Amount of code needs to be proportional to amount of layers
    - Slow down of core will push velocity to outside repos
  - Take things successful in ecosystem and adapt/migrate to the lower layers
    - Kind of like incubation process
      - Incubate is too much to the side
    - Identify extensions
- Customer anecdotes
  - A ton of churn every month
  - New pieces and patterns pop up every month
  - Need to center on “blessed” pieces
  - Potential effort to slow things down
- Repo level refactoring
  - Not make frequently
  - Easy to pull in new things instead of tossing out