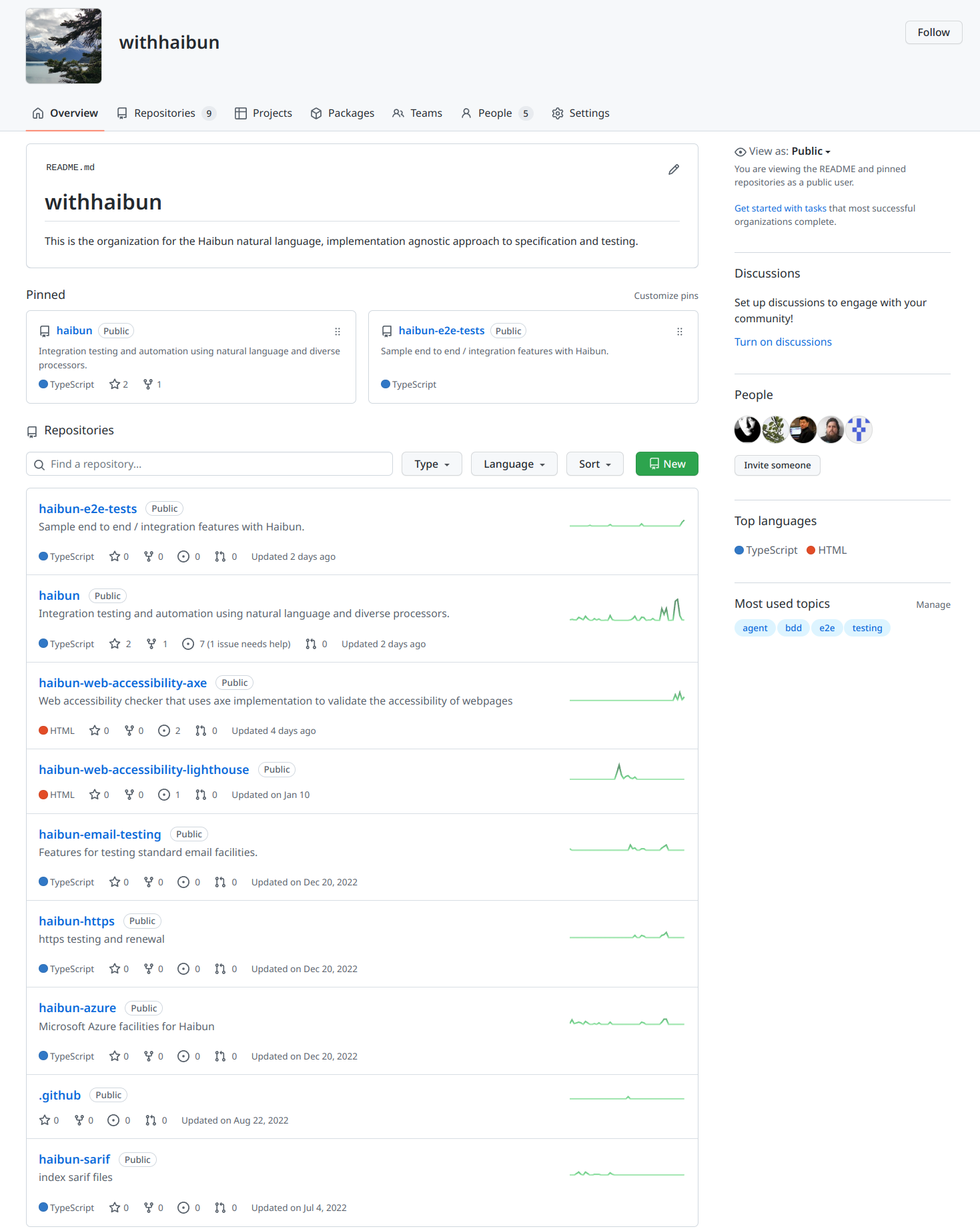
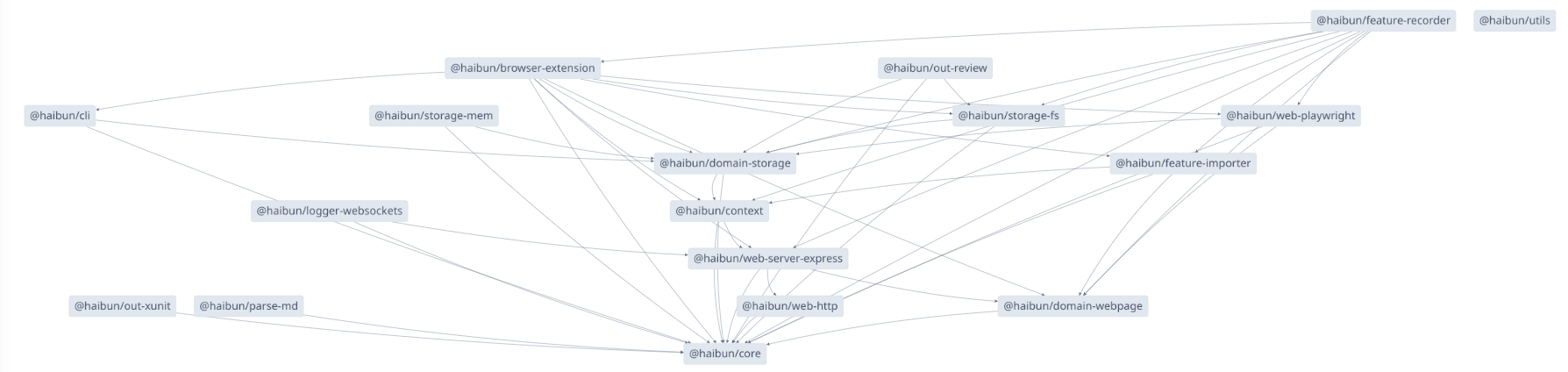


[*Overview diagram*](https://viewer.diagrams.net/?tags=%7B%7D&highlight=0000ff&edit=_blank&layers=1&nav=1&title=planning#R7V1Zd9s2Fv41Pqd9sA53Uo%2Fy1mYaT9M4J2nnZQ5FQhIbitCQkCX11w8AggsWSZTFRY7tpI0Jghvu%2FS4u7oYr83a5%2FSX1V4tHGIL4ytDC7ZV5d2UYumO5%2BB%2FSsstbDNMY5y3zNArzNq1qeIr%2BAezSonUdhSBjbXkTgjBG0YpvDGCSgABxbX6awg3fbQbjkGtY%2BXMgNTwFfiy3fotCtMhbPcOp2n8F0XxRPFl32PdN%2FeD7PIXrhD0vgQnIzyz94jbsG7OFH8JNrcm8vzJvUwhR%2FttyewtiMq78iD3sOVu%2BcgoS1OSC3%2F6wd8%2FJ31kC4q%2BuPY42u1%2Fvrg3Ly%2B%2Fz7MdrNhjsddGuGB36fYDcRr8ybzaLCIGnlR%2BQsxvMELhtgZYxOz2DCWIExi%2BEj6M4voUxTOm9TKCHNnBxe4ZS%2BB0UZ%2Bi4mTfyN7HPfAYpAttaE%2FvGXwBcApTucJeSG9l4M1689tjxpiKsq7G2RY2otssafcZM8%2FLe1ajiX9jAqgf5z9%2FWm%2Flj8ut8%2Bvyf3%2F%2FM%2Fv2vT4l9bSmG2InxY28mTwisViDFx3NEvx9znKmTcfsGpter2N9tUvqKIk0wN63Ir0sYrknbjR9H8wQ3xGBG7pRhAkXJ%2FCM9yilR9Ajw4OJHmjdkUCMMggk7geCKJ6BuyAQMfeDNAomA%2BIwTeGA6K6kokUxB2L1UNDSdo6Jhy1TULUemotMCEdVIsRsgpYaDYnA%2F%2BlMQf4JZhCJIBnkKEYJLFTli0vOmFCi1kZ3Rn70UyylRCC5DRRtN87QHwgSlFCIHoZ8tKK7ZmZyjtnMi5UcRzNxRhCVuNgpiuA4V2G4Brobm8HAt5WWN0oanwKtudUVqXbclUt%2FC5QpmeLAMDc4kuoMQTyfsEKZoAecw8eP7qvWmkqFk4Ko%2BHyEhIeWYvwFCOza8%2FhpBXq7iIU53f5LrR6Y3Lhr%2Bqp%2B927Lb50e7%2BtEnkEZ4dAij3R0kXQbXacA%2Biwku5KdzgA4OmJn3JONwkOopiH0UPfPTroqA9NJJmvq7WocVjBKU1e78iTTUmEmQ%2FbZlC8yQ37FijfLVGnGLLMifMHr9FENFYgnVNHlUAi%2BjMKTsQpHGbqZLEDdlsTyzyR92Za09%2F1GKa%2FrTEop1ixt4RwHisQLDnUlrXaHXfEgy5Cco8hEYAsDivAq2EcrxbFkuOy7hjH%2Bv0EwOdrWDF2F5zzB1A1wJmZYwn5tj4Ra5jGFXCcQ%2FDaTq2dtUifQE%2BQGSoTsAM2DOTNGErGQqZZi2PUTkQ88lczH6xyW52TpDnIfj4s1rdLtPyCy8zojavEcrXi%2FjSYBgXcTu1cPUOlWdVo1WMPjM2HFN35Gk9%2Fni1fP4ec1VaMNmn%2BK1fFqNLA%2BpvwSDgKlUjcqDv6hctY8oRn2D0GgKQsO%2BKBAahbWjRu3P4DkCm2HJrXVPbnHxawMvtFTY94ypeVibapNBxpfFIOWb1xjkDjyDGOa2DWrrmNasHAOI7XNI11xSu8Jy1nUHltS6rli0ILDKqJlUnkF7wO8ZCOlonXlUXbWsztRVpcVQVlbvtyBYI2p%2FmO4uSOoekLlmKZDVCxaBiGeyxHGhma91%2BrZJ6MwGUdokdMFMf6Q%2FVvYEDjvPhnF4FOv6HPDROgWU0fCSeQkSxVLpbVk5TE1JymNWDrsz4V6sal%2Bnvn0G6gsNuY5667J0IkMW4ncgi1Iqw2frJCAaDcYKIu%2F005e7u58lWr7U8XaW3%2BYsiFiFICkMgY7Cb2P0ulR13CEw0vPasvBUd60YmTovAh1Rtu1RjE6dNS1X%2FZxuZ8FiFLl17jwFWUYWHz8COoUJbHB06q5sWsCIeN12pDNwbDbWZl23E8SfClTPcHmWcnsAquHI3p0BDVIXb1hq36ivZoaxqCCLcqNjL0%2F5oTW%2B%2BGOCb%2FW0Xq0wYd%2BUKcrTDZ4YikCofp0GmqwRf0rhcxQO4o99JV4D05VnhAtb6RSvWDcxrkAQzTAoCFaIrREDzl9S6wD9P255Qn4S%2BmmYNUBlm6rWbDYzAqWqFTpTx24JfbbgslNFRPS8EHJlI%2F0lwK8HL84Z8LOMpgpZ%2B7bpMyNgZHI%2FxFChIdWIokJWMSNGGbxdTwEZ8yrq%2Bfc1iqOksN%2FhLpNknpvzbAWthBjV09dAC5hG%2F5CwjaO%2BhBMWRTxQLc%2BWgKoy6ulaZ2siW3bZ3C8jOQZ4QJS6PEz1hlGHLa6UrOZ2%2F9adpS8zaQiLb8M97AiQ%2But9OAKGsZbVglttq85p%2BkjTnCMzwuVwmtONFe70MFg%2BGrOPJXk5Sm8sy%2BG8XJUiCrKAuKVQEnvNciiFpYKMHz7C%2BVxJxfzENSbmEwy%2BA4Wf8AcnpCkoEboi7KNnQtrGQUGOvzVCu89UDsGkVWGu0shBEoo6Opgv8wGnLBSs0%2BdygZfL5FKkP%2BXMwd6I2ZDUJ9sU%2Bz25VMSkBbOAYF9GM0v2etBsl4R4%2FEd%2FN3P61%2BjPL9b9NGAEsRuhugwO4IMB6K0wq2WMuJXMwEdEZUApAA8wDvMWnbU84vVt2ZCATY3R765cIsP%2BtyYJmDcVNsqmK3NSO78HL1Vv47Y6qEFAfbs8T1R5bQmVk68sIFW%2FkCiRtS4MZlwP%2Fh5q6NUv4O%2B4B478Be6dAptl9pnCRDqbAWePkcYdT7US6oyURhuTseByNBX2UqXFxuxMhivcHENo51LmSi0VzeWspy9W1%2FfSr%2FtIP3OkO%2Fa4%2FOFVZ0zckek0kseKW495hnK1EV7UGY5r2LbteJbF35dB72zXufDYgj%2B79cgpol1uU%2BCjgT2519pIM7hsSW2k68dsjI1WlB0bHu3GKTt2657gMzVPRcoOjV4l4cZHTf34CdEqAzXD43oZk1StFOLXeBVhFqY4kxR2Aa4EgTyReJ2Z%2FguP0XBWHc%2FkQWj0hkFRs%2FACoNYspp5t2eeiVhGXqCaJ7nUywZ06W%2BCJiONV60h48pH%2BXc0usodxgpAfLGgwJea7gbnbcevcTSadquWS55im3Hop0UaO6fHcZx3m1iP9OzKF2vttaNNi3vsMsnVcBqTg50z3zojZJlrGPqV9fc1DrSaLKA4%2F%2Bju4JtTDnBF8L45kXxllnKJ8h3NgcqyufCJ3ZM9KAVl7fyr4TBeaHv0t1%2FGjn6HiLWEc%2B6ssmpaB%2FkvMcFFywwJoaKeD8VQdTta24GYxNZXpTukAtPZz95kalGyR%2BeRnJIyCqFDazI9iiUvw5yJprJjIqBOaNQn21z2FZTirrFW1fGbDQJogvnYWUwm1iMIQJFS4Ih%2F505JPGa7xi9o3%2BC%2BxIBBPv31HTAn2jV4d23e0e4qwyoe%2FxY8oYQFmpg0gDHWqAihzyBHMHmecYl0lK3VKPjGczthEnhW%2F4OnknTWGZo3C9j4ga8iWo1t%2FRfK55PX4O3v0zB6uNzR7OLKz9huYyqvzyg8gcc2lRA01sBS3oCWIRQUVjtp%2Bg4QMW06cKOhHBpAjVmGLJyeucw%2FKBHfQjdWWDk%2FlTcjJfgdWIAmpvpHUVNT8vvvsNr3aoamHuOPVUWG1Ob6WvzALnOHoQ5BnyLocZ9j12mSEQkwMHWqmCeEIRyLNxO69eAxcOcqRlzuDmnQq%2B2STEos9221OkEytBzGcqXjIK9vJahWzRAGJ5rQk6nHVLVrm9Z2j5Ry%2FVhBHeFQxfz%2F8Avz0v7rhbfF%2FoxWJA7iM5ADd411016rkAJVG2GEepiYN%2Fnu40EmYHHci%2BhuHEfcWLqQoBvY7FttfQAwqXnwPF3oPF%2Fqxw4XECgXK2F1VfmV34ULeYRneldJU1LG9qmrYMqV%2FfxXb4TPtXUVeZa%2B1NU721AoMZ4%2F78GW5so35Zj2nH5tnVAsM9wozNcVxVQG536IYnjzFEshKg10Y2RZg689JTvrNquZQZq2lrDUaECTagmJ7losJpxG1ZV3hoXMVsfXd7Q9QmPq4oiUpmMHtmyUR72fXbXNoEslBaI9wGsVvFURiTlipmgxFIaMwBbzXauEZt69aLWKQoqiXdr2OHO9PGKxFycTAz0gNym9%2B%2FB0tMPnni8GCZkztaqDgmHP1qD5cZGPBimwpJIw6kMbobhaQbcwkQoI6Hw3taxQCOXSwXXd4ZZ94qUe8RQ%2B46FzvyCNeIruxR7wpq3S3dh7LMXtPIH2OAhIwoS3x0LzzyfB8oqpX2DOfyA6MOzxW7zxyMTwyHppHDE1evrNCSwPvlzHinJqVee1iI9QNja0rG1R5MTrRm0%2Buh8lXby92I93nej%2FcvSPPuyYHB3JOWC2ESwx4WaeuErLoZj%2FHUU9HC6T3zyAXOyoOOl1ZlZxCLWivNq%2B8qnaNtW2V8trhHkvyAulDEtH96151edyut1lqXK6n%2FW2WXiYyxFKB4yMy43D%2FjoSGKS%2BlPhD4vaqqu72wVbF3zNBsNebnFtfohU1UJdQvYKuwkVMXWvpIM80jYqurBNSO9xBrzKbjCwsN0%2BTpDmvN4TogPI1VZzznv7HNxMZjQdAPvZkY1nYU6GYmWoEMP2jonmjlNBS5%2Ff2G7hmKXSsuofQGKXbCpUX3tt7sM%2BO%2FzOMfIof6TDDLy8AHVWbpSSEdVfye9TJSNA7xOSFUS1CYraa1uDt0fg6ycdelay9GQYSunZ%2BeMLOaWrMaT%2B0VtJAjSPIM7%2B6w14ByHWBP3BDV8BTzZb%2FYK9nxHXuqbPLjSSXto%2FQ8ghabs0p7xJPCMBrIfwUZooHGWpjiN6JWznyH4jwGWiP2Ts2fEl9HkO%2BzCpLnKIWJeuNL5tnh8hhOc%2FIcD2lXwZ3nQzEjsh7qYLWCXg68rsL%2FZjq9%2BlZsWXBO9hCtvlY8Ixn2p9sYrkMaxK5tYPqd7IZKzeSk4etj%2Fi9Awc%2BqtWnekq38ZP87bNjAkbewWDSG%2BBLyvSeUXzmzPVr4hJPXGfVZo0UK6O6tpRv7Vr4LzQ7Q4Ix82yIKFoSOPt0bmngOCUZmURKxTVzIh%2B67RZTRDyRbusR5yoHQzacwrHXGAAIplqD415Hc%2FcsC8H3YZ3AvqNg6ZorXovmDyvfJt5jJ951JptlKOZw%2FASIbbidfng7QEbNsTso9pofzhUJeBr0tmZAvk7lNds%2BXCXaxj0zhKlEEbyjL23UX52MU4aY%2Fcj660bwi5IVYoHVN2Mva0%2FowQSvqT8rcUVaZrIOVWQ7zmD%2By6bWsg2n0R6WDzWbsjMgWLWBO14Sd9VxFlEOvyUaGK5trD%2BzoxcTy61C7X1rI4LTKxg0A3zTNu6vgB3n73z2pMi3v%2Fis85looSHD0tcwedsoxFEXangod5W3YwksZUNKpaXxed8ZwRU20N0YVS9iLZPjiAuWd3y5RbHGDmKZVJjskyjA1oYtsYbaFXJEwfNUoXVgbjcdCIWnNfQ3V3MfN3fKXtY1kuZKrexcXKVSUmXwriXT8hufFHNh%2BHh0%2BTCGxAlVqCx6ixSMMiYS8%2Fz8%3D)



*https://github.com/withhaibun*



*Module dependencies*

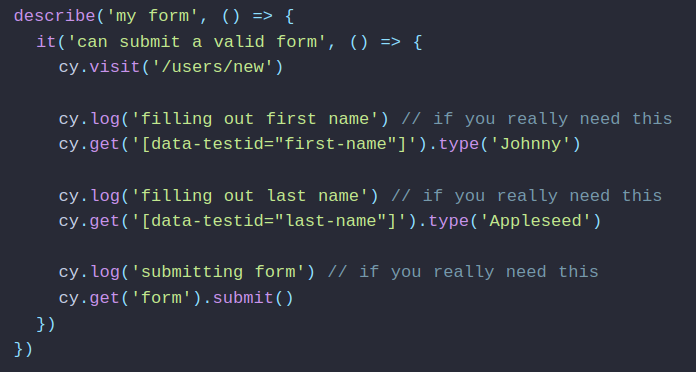
| Essential *modules including abstractions; domains, storage implementations* | *Module ecosystem* |
| --- | --- |

|  | trigger:  - test-e2e  jobs:  - job: e2etests  pool:  vmImage: 'ubuntu-latest'  container:  image: mcr.microsoft.com/playwright:focal  volumes: [e2e-reviews]  steps:  - task: Npm@1  inputs:  command: 'install'  displayName: 'installing dependencies'  - task: CmdLine@2  displayName: 'Running login e2e tests'  continueOnError: true  inputs:  script: |  npm run  HAIBUN\_TITLE="Login e2e tests" HAIBUN\_DEST=e2e HAIBUN\_O\_OUTREVIEWS\_TRACE\_STORAGE=StorageFS HAIBUN\_O\_WEBPLAYWRIGHT\_STORAGE=StorageFS HAIBUN\_ENVC=${\_ENVC} npm run test-pipeline-login  - task: CmdLine@2  displayName: 'Running regression e2e tests'  continueOnError: true  inputs:  script: |  HAIBUN\_TITLE="Regression e2e tests" HAIBUN\_DEST=regressions HAIBUN\_O\_OUTREVIEWS\_TRACE\_STORAGE=StorageFS HAIBUN\_O\_WEBPLAYWRIGHT\_STORAGE=StorageFS HAIBUN\_ENVC=${\_ENVC} npm run test-pipeline-regressions  - task: CmdLine@2  displayName: 'Generate sarif indexes'  continueOnError: true  inputs:  script: |  HAIBUN\_DEST=sarif HAIBUN\_O\_AZURESTORAGEBLOB\_DESTINATION='staticanalysis-sc' HAIBUN\_O\_SARIF\_TRACE\_STORAGE=AzureStorageBlob HAIBUN\_O\_SARIF\_INDEX\_STORAGE=StorageFS HAIBUN\_O\_AZURESTORAGEBLOB\_ACCOUNT=${\_O\_AZURESTORAGEBLOB\_ACCOUNT} HAIBUN\_O\_AZURESTORAGEBLOB\_KEY=${\_O\_AZURESTORAGEBLOB\_KEY} npm run test-pipeline-index-sarif  - task: CmdLine@2  displayName: 'Generate test result indexes'  continueOnError: true  inputs:  script: |  HAIBUN\_O\_OUTREVIEWS\_TRACE\_STORAGE=StorageFS HAIBUN\_O\_OUTREVIEWS\_PUBLISH\_STORAGE=AzureStorageBlob HAIBUN\_O\_OUTREVIEWS\_INDEX\_STORAGE=StorageFS HAIBUN\_O\_OUTREVIEWS\_REVIEWS\_STORAGE=AzureStorageBlob HAIBUN\_O\_AZURESTORAGEBLOB\_DESTINATION=e2e-reviews-sc HAIBUN\_O\_AZURESTORAGEBLOB\_ACCOUNT=${\_O\_AZURESTORAGEBLOB\_ACCOUNT} HAIBUN\_O\_AZURESTORAGEBLOB\_KEY=${\_O\_AZURESTORAGEBLOB\_KEY} HAIBUN\_O\_OUTREVIEWS\_URI\_ARGS=${\_O\_OUTREVIEWS\_URI\_ARGS} npm run test-pipeline-indexes |
| --- | --- |

*Storage implementation and* [*pipeline*](https://raw.githubusercontent.com/vid/gccollab-e2e-tests/main/azure-pipelines.yml) *for Azure (files & blobs). The pipeline uses Azure volumes to store artifacts with retention schemes per test or deployment type.*

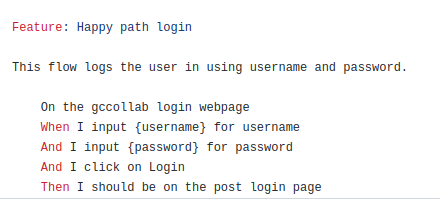
* Implemented with minimum of dependences
* Highest fidelity for testing in each domain. Generate diagrams, videos, etc.  
  Application tests only contain the minimum versioned high level language, that could be understood by anyone.
* Very high level of reusability between systems, reusable, versioned flows can be based on specifications, standards, components, applications
* Scenarios as descriptions of swappable systems (identity provider, cloud, services, interface)
* Very practical yet very wide potential scope
* Seamless integration in best practice system
* Based on very standard approaches (Playwright, Jest, Testing library), possible to “eject” basic standalone tests.

Compare to current “standard,” cypress;

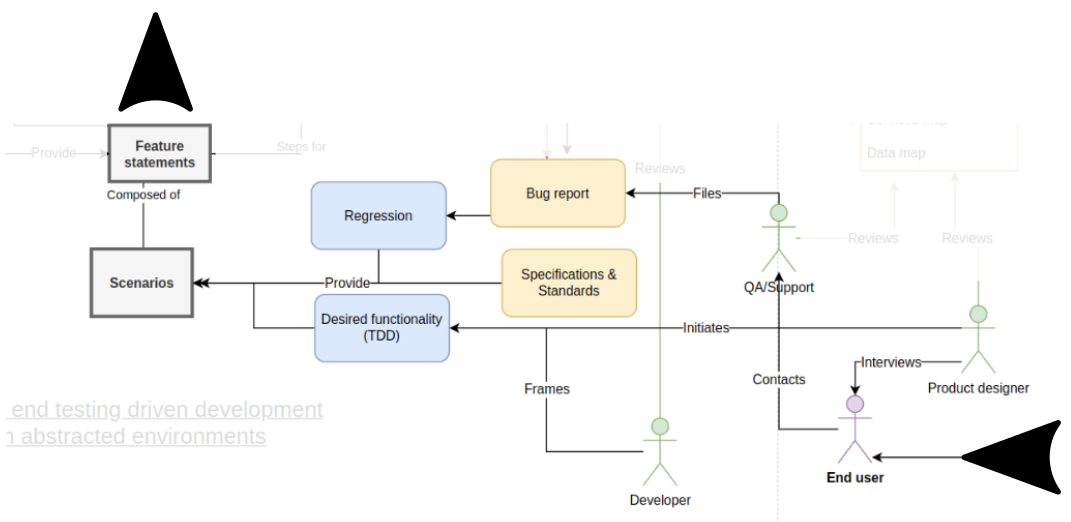


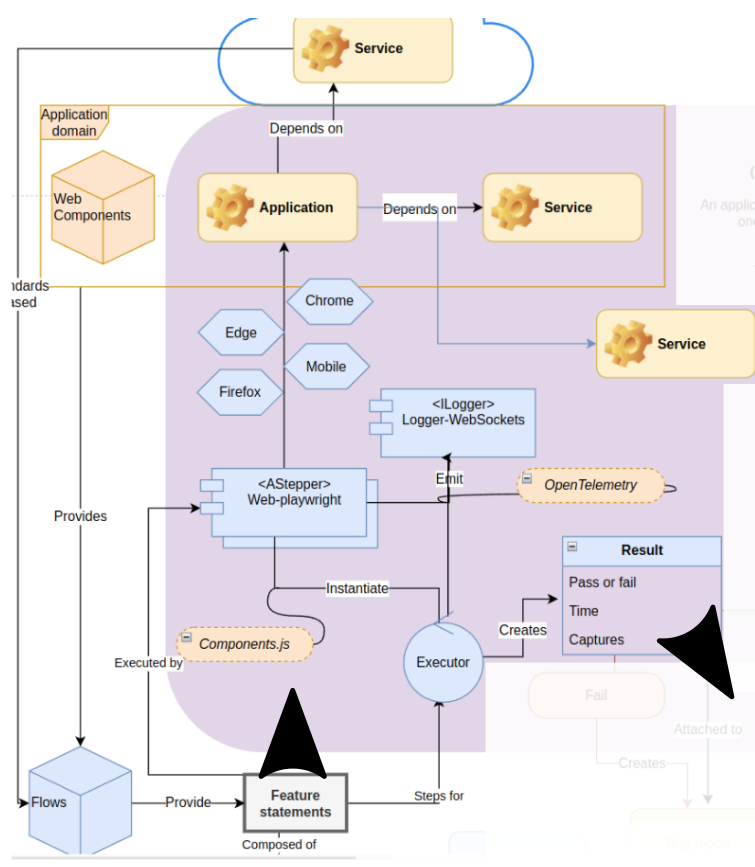
*Cyprus test*

* Completely tied to specific implementation (not abstract)
* No consideration for connected services (not comprehensive)
* Gobbledygook

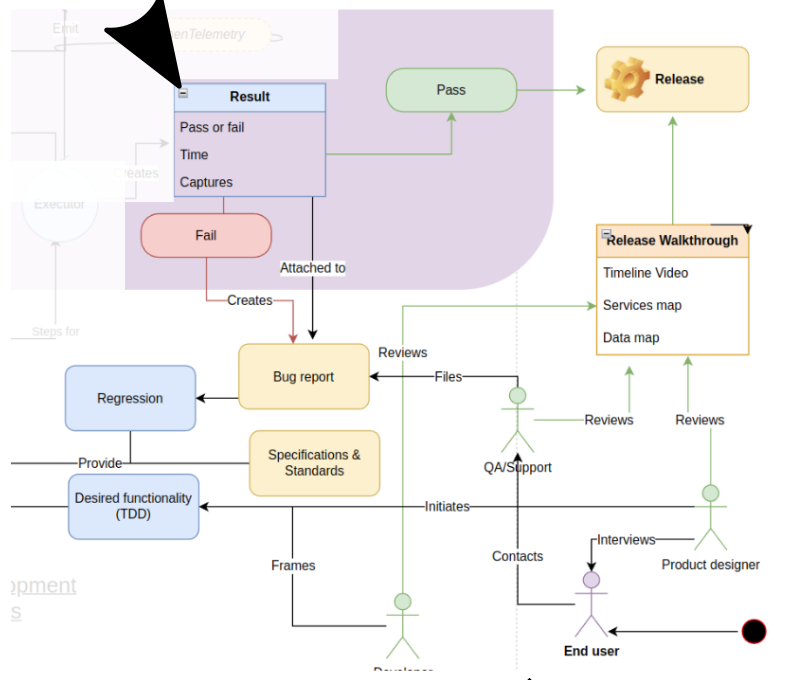


*Haibun test*

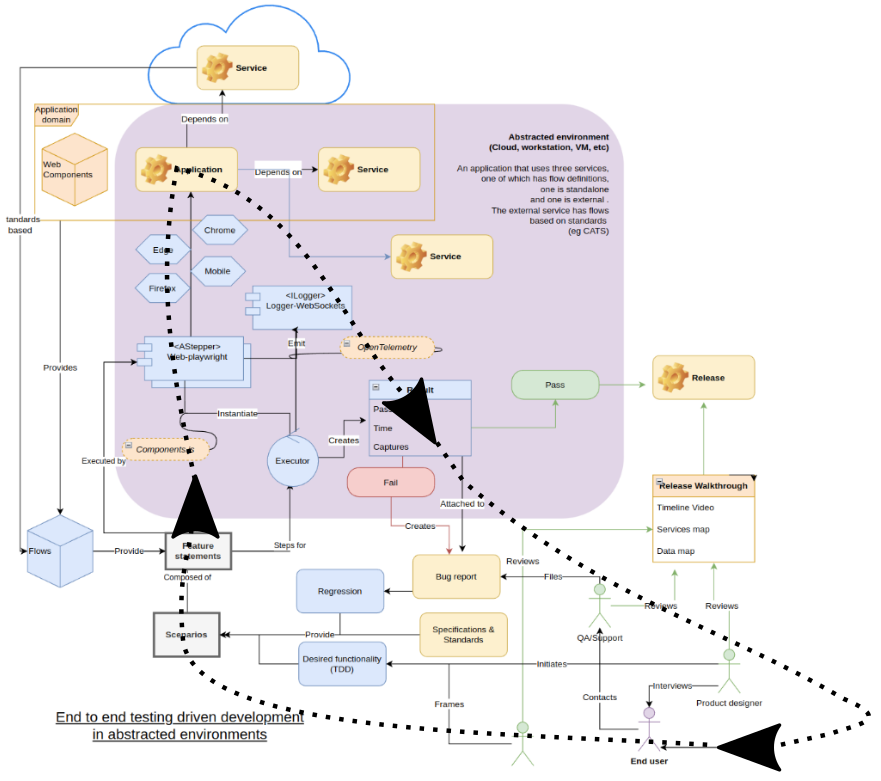
*Workflow for specification, test development*



*Abstract evaluation environment (cloud, workstation, VM, etc) using features & flows*



*Results feeding back into workflow with deep, interactive artifacts*



## From here

* ➡️Add mapping
* Use OpenTelemetry for messaging
* Improve reviews
* Alert for unexpected service changes
* “View source” (with versioning) of any command
* Adopt components.js for component model
* Abstractly automate connection with Azure DevOps
* Improve browser capture
* Editor helper
* High fidelity load testing
* Save, resume, branch sessions
* Automate security process and verification
* Specification as test as documentation as data map as application generator (OpenAPI for services & components)