

Meeting Results of Interviews, conducted by Jamie Thompson for Work Packages C, D

Interview with Martin Odersky 9 Jan 2026

- We should revive the scala/toolkit library and add new components to it
- Jamie to post on forum to gather feedback
- Python Generator style abstraction
- Default Sequence collection type
- Explore data science type libraries
 - Tensors/matrices
 - Columnar operations, e.g. for spreadsheet style calculations/big data ops
 - Using Named Tuple types to achieve safety
- Can we improve lihaoyi/cask to be a more useful framework as a server library for toolkit
- More “derived” instances in library
- Missing “CanEqual” instances
- New Result type
- New Validation library
- Should we do a standardisation of JSON + schemas?
- More useful operations for NamedTuple

Interview with Seth Tisue 20 Jan 2026

- Are there too many collection methods?
- advice for making a roadmap:
 - collect a list of suggestion areas
 - inquire which is most important
- We could separate more of the old/unused packages
 - deprecating
 - prefers investment in the newer parts of the library e.g. Scala 3 stuff
 - e.g. metaprogramming + compiletime ops
 - MacroAnnotation
 - On scala toolkit:
 - It's fine to be opinionated.
 - Scala-Yaml?
 - JSON?
 - Ok to drop Scala 2 whenever
 - If so then make new major version of toolkit
 - could still have security patches for old Major
 - Let's add new things, and enhance what's there
 - keep stuff up to date
 - documentation + publicity
 - parallel collections why not add it, to Seth it isn't so important that the threading is opaque for the simple use-case.
- When to do benchmarking E.g. when we replace a collection
- Seth thinks default Seq should be Vector

- Chunk?
 - competing designs from the ecosystem, perhaps too specialised?
- publish micro-libraries that can be candidates for inclusion
 - e.g. Result
- Its a result even if all we do is "write down the situation" about various things
 - what are shortcomings
 - leading proposals
 - plausible path
 - "even to document well the reason why we do nothing for X"
- pick some people to ask directly 1:1
- scala-only json? (don't care if the schema is compatible with other platforms)
- TUI - talked about it
 - e.g. for blog post for CLI post/guide - list all options
- Better scala android + iOS? guides
- Document to explain diff between Ox and Gears and discuss the possible futures.
- SIP is important because it forces ideas to be concrete
 - useful as a model for any proposal for a library.
 - key for consensus and planning ideas + possibly.

Interview with Li Haoyi 30.01.2026

- Benchmark collections
 - <https://github.com/lihaoyi/scala-bench/blob/master/bench/src/main/scala/bench/MemoryMain.scala>
 - Strip out the old sys and io lib
- Modularise or modernise - The old scala.concurrent
- Annoying that mapValues returns View
- In Haoyis view - Scala could promote some com-lihaoyi libraries to the core:
 - geny.Generator
 - Easier to implement than iterator
 - Less powerful
 - Push based api
 - PPrint, Sourcecode
 - Stability is important to Haoyi
- Toolkit libraries
 - Focus only on JVM?
 - uPickle, os-lib, substitute requests instead of sttp
 - ScalaSQL (fleshing out)
 - Websocket Clients
 - Cask (but first it needs fleshing out)
 - Needs production ready inspiration from the big frameworks
 - Django, rails, spring boot
 - CSRF, OAuth, Concurrency
 - Forms
 - Scala CLI should include Toolkit by default, and with explicit opt out
 - Sbt possibly also include Toolkit by default
 - Don't bother to add a TUI helper lib
- Scala.js needs more care and attention
 - e.g. DOM API wrappers
 - Integration of backend and frontend
 - Node FS wrappers not worth it
 - Not unless you are an expert.

- Haoyi suggests that the amount of work into Flutter and Jetpack Compose is a careers worth of work (on topic of a comprehensive framework)
- Can we use more of the static information of Scala used by Scala
 - How can we get it to stop the agents using grep all the time and not using our MCP tools
 - SKILLS?
 - We should invest in that.
- Collections
 - Don't bother with Heap, Java priority queue exists.
 - We could see what's missing in java collections
 - Hard right now to turn an iterator into a sorted collection
 - Via a heap e.g. and not copying twice
 - Better sorting
 - Should we pull out the concurrent.TrieMap?
 - <http://lampwww.epfl.ch/~prokopec/ctries-snapshot.pdf>
- On DataFrame - have to invest properly if you want to work on something like a Pandas equivalent
 - Or find a partner to give you feedback
 - Fast compile times
- Can we make it happen to get compiler performance improvements?
- Scala's reputation is fixable, we should go and tell the world:
 - Devovx, java user groups,
 - Scripts, automations
 - Scala compile times a problem
 - Python conferences
 - Concurrency + performance
 - Compiling is paramount
- Clean up the Scala website
 - Hands-on Scala front and center
 - Consolidate
- Getting started guide
 - Too many install steps
 - SDKMAN no
 - Bootstrap script in the repo (just cut out coursier)
 - investigate
 - Push for 1 tool on getting started
- Need someone else to write a modern hands-on scala.js and keep it separate from toolkit docs
- Everything needs to be real high quality to explore new things
 - "Graveyard of abandoned Phd projects"
 - Small scope and complete
- JVM sovereignty
 - "Would be nice to have Scala-native standalone"
 - Scala.js amazing compiler but ecosystem lags
- Needs design partners for the concurrency platform
- Scala-lib is minimal, toolkit is convenient.
- Os-lib does support other filesystems like the in-memory FS

Interview with Tomasz Godzik 30.01.2026

- Stability of the Toolkit is very important according to VL.

- We could do a toolkit-experimental for incubation.
 - Need a good reason to remove a lib
- Agree that we can be faster to change things
- What about “lost media” if we change something
 - Candidate should be stable and works
 - Preferably with maintainer available
- Concern about the maintenance burden of “adopting” libraries (i.e. taking over maintenance)
 - This is extra concerning because that suggests we can't do anything at all because someone will need to maintain it.