



sans-io: meh.



future james here:
time travel slides
were a terrible idea.



sans-io?



use normal, blocking
functions



do your i/o elsewhere



it seems compelling



async: let the
compiler make your
state machines



sans-io: hand build
your state machines



async fns:
hierarchical state
machines



surprise:

james has walked into
a trap



surprise:

james has walked into
a ~~trap~~

spirited debate



opinion: async is
great for protocols



counterpoint: current
i/o abstractions are
imperfect



opinion: you should
build your own i/o
abstractions



opinion: sans-io is
throwing the baby out
with the bath water



counterpoint:
improvements still
needed for async



(tracing and tokiyo-
console are very
cool)



admission: there is
still forethought
required for success



counterpoint: async
code is not the same
as blocking code



(n)rvo:

(named) return value
optimization



counterpoint: async
code is not the same
as blocking code



counterpoint: rust
needs (stable)
generators

Self-Directed Research



(deep breath)



admission: there is a
step from rust to
async rust



opinion: you can do
things with async you
can't (easily) do
otherwise



reminder: this isn't
an actual fight



admission: there is
still forethought
required for success



opinion: you
shouldn't treat async
the same as blocking
code



opinion: async is a
good tool for state
machines over time



opinion: sans-io
would be much better
with generators



opinion: I don't
think I would enjoy
writing protocols
without async



see you in season
three :)