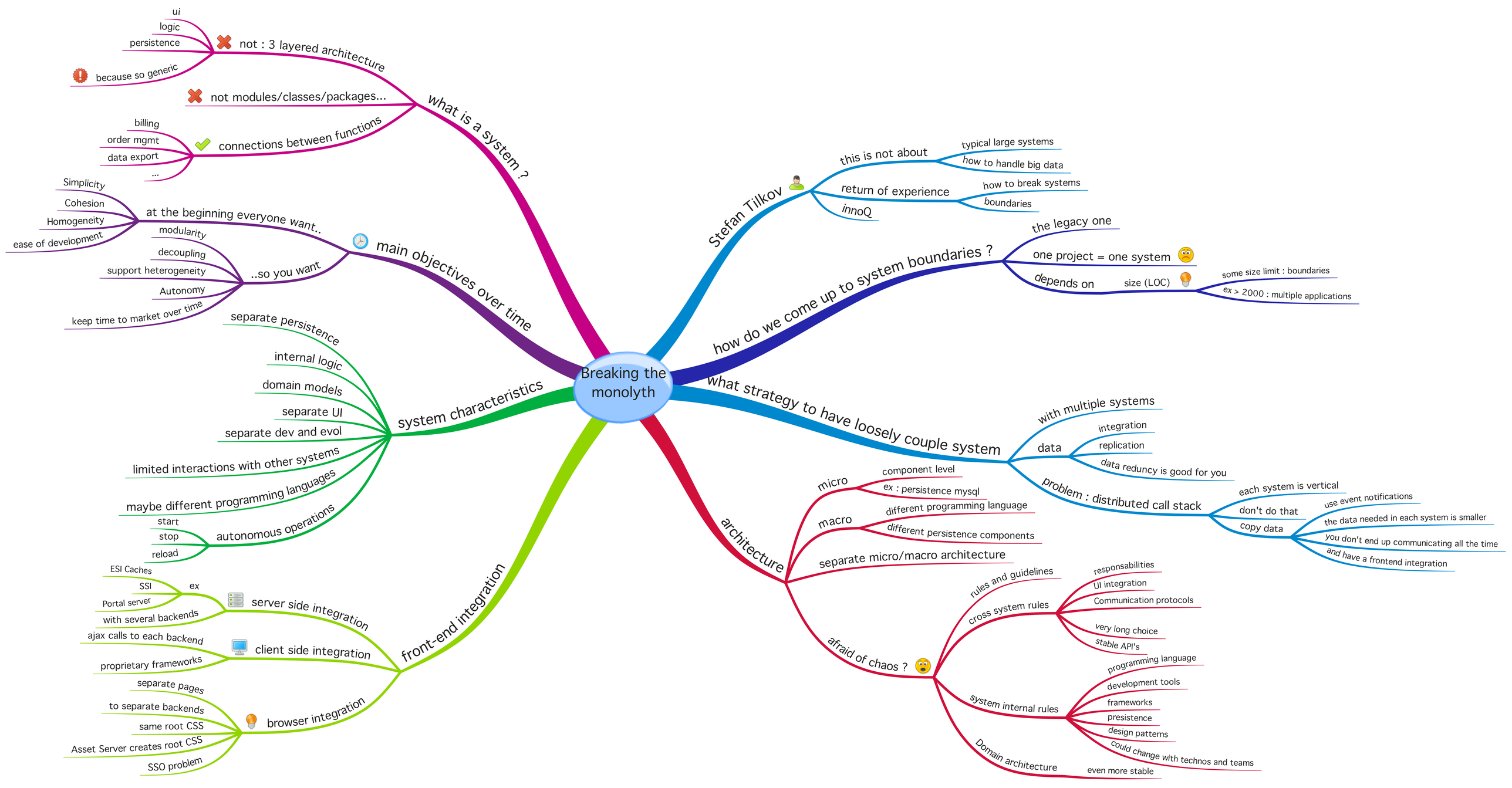
Breaking the monolyth
what is a system ?
not : 3 layered architecture
ui
logic
persistence
because so generic
not modules/classes/packages...
connections between functions
billing
order mgmt
data export
...
main objectives over time
at the beginning everyone want..
Simplicity
Cohesion
Homogeneity
ease of development
..so you want
modularity
decoupling
support heterogeneity
Autonomy
keep time to market over time
system characteristics
separate persistence
internal logic
domain models
separate UI
separate dev and evol
limited interactions with other systems
maybe different programming languages
autonomous operations
start
stop
reload
front-end integration
server side integration
ex
ESI Caches
SSI
Portal server
with several backends
client side integration
ajax calls to each backend
proprietary frameworks
browser integration
separate pages
to separate backends
same root CSS
Asset Server creates root CSS
SSO problem
Stefan Tilkov
this is not about
typical large systems
how to handle big data
return of experience
how to break systems
boundaries
innoQ
how do we come up to system boundaries ?
the legacy one
one project = one system
depends on
size (LOC)
some size limit : boundaries
ex > 2000 : multiple applications
what strategy to have loosely couple system
with multiple systems
data
integration
replication
data reduncy is good for you
problem : distributed call stack
each system is vertical
don't do that
copy data
use event notifications
the data needed in each system is smaller
you don't end up communicating all the time
and have a frontend integration
architecture
micro
component level
ex : persistence mysql
macro
different programming language
different persistence components
separate micro/macro architecture
afraid of chaos ?
rules and guidelines
cross system rules
responsabilities
UI integration
Communication protocols
very long choice
stable API's
system internal rules
programming language
development tools
frameworks
presistence
design patterns
could change with technos and teams
Domain architecture
even more stable