

# Is your Automation Infrastructure Well Architected'?

Adam Goucher  
CTO, MobileXCo  
@adamgoucher

[Browser based] Automation was largely solved 6 years ago. The *infrastructure* and *operational* procedures around the automation environments remain a huge business risk.

# Your Automation Infrastructure

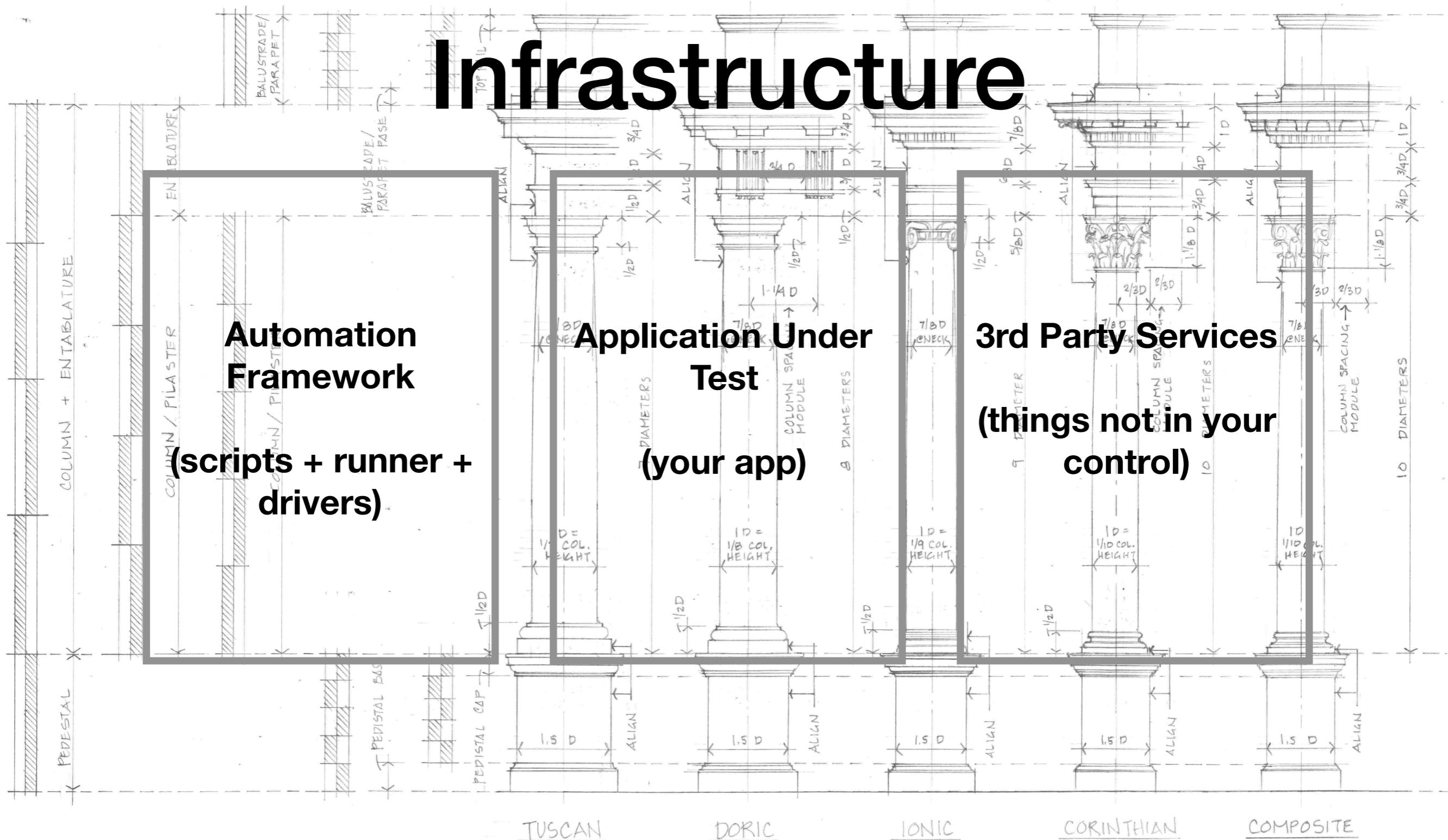
**Automation Framework**

(scripts + runner + drivers)

**Application Under Test**

(your app)

**3rd Party Services**  
**(things not in your control)**



The ‘Well Architected Framework’ was designed by Amazon, but is [at a high level] cloud neutral.



# The Five Pillars

# Operational Excellence



# Perform operations as code

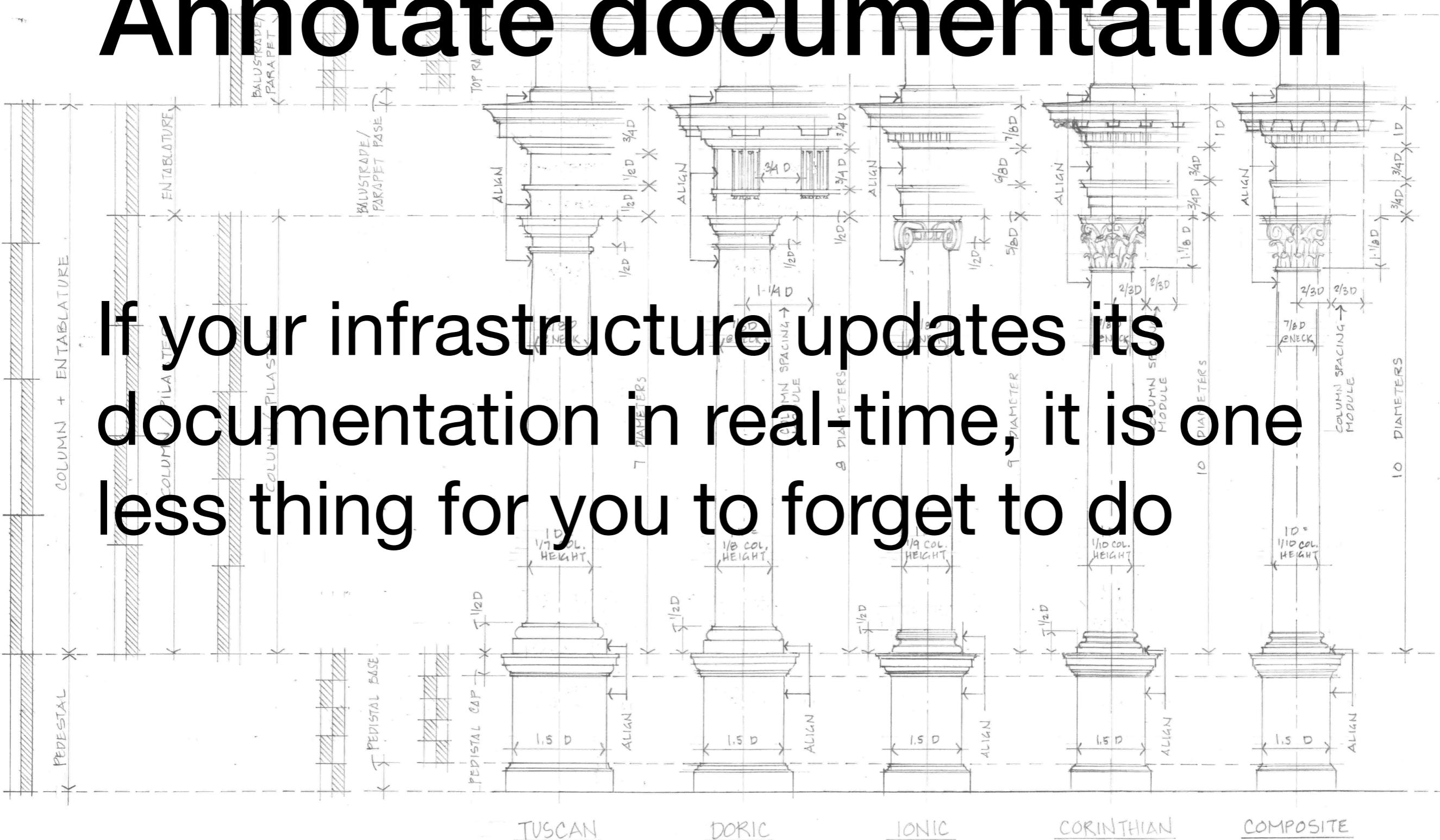
Every time you provision or configure a piece of cloud infrastructure by hand, you are doing something wrong.

Every time you log into a box to do something, you have found something you need to convert to code.



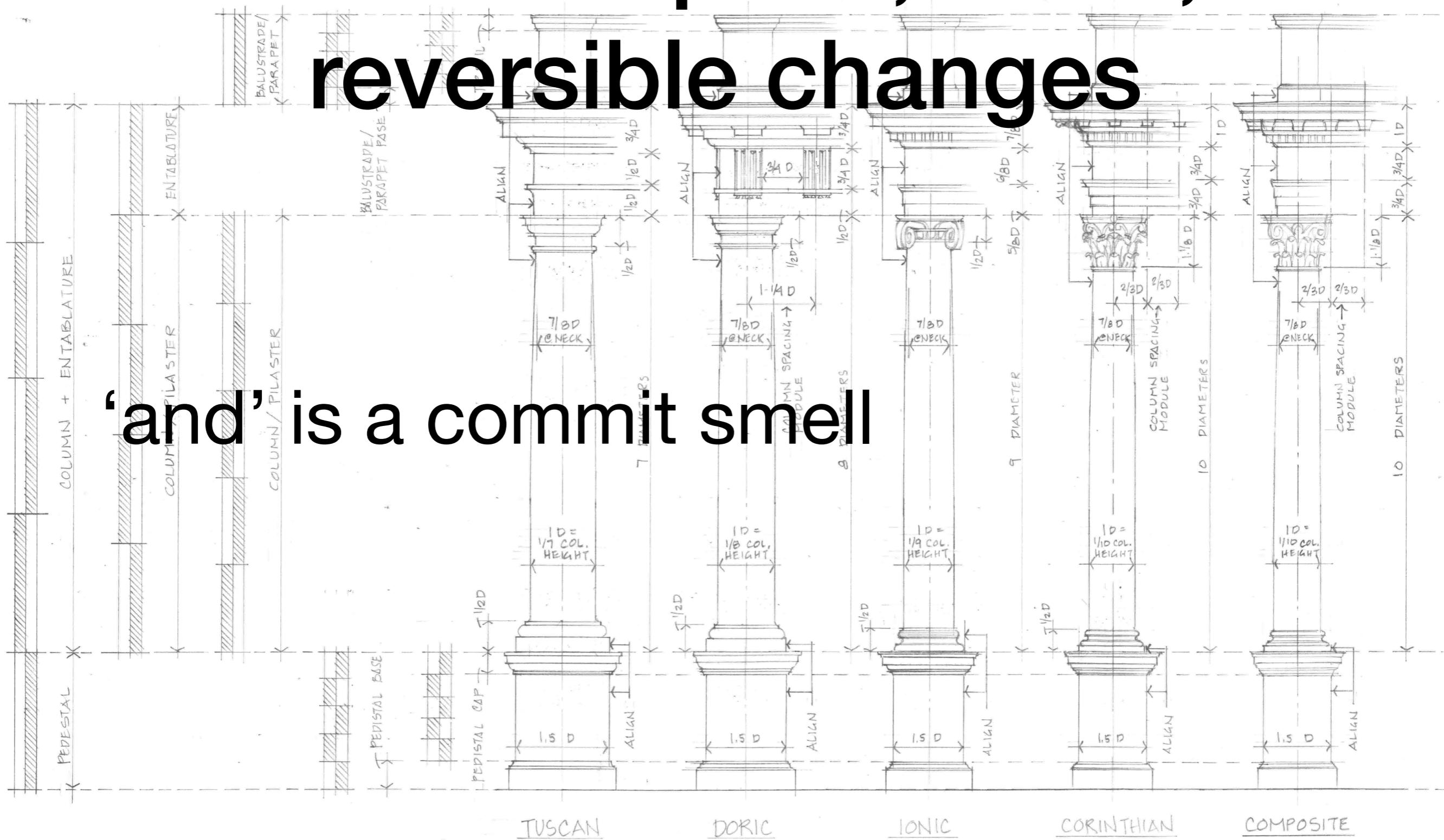
# Annotate documentation

If your infrastructure updates its documentation in real-time, it is one less thing for you to forget to do



# Make frequent, small, reversible changes

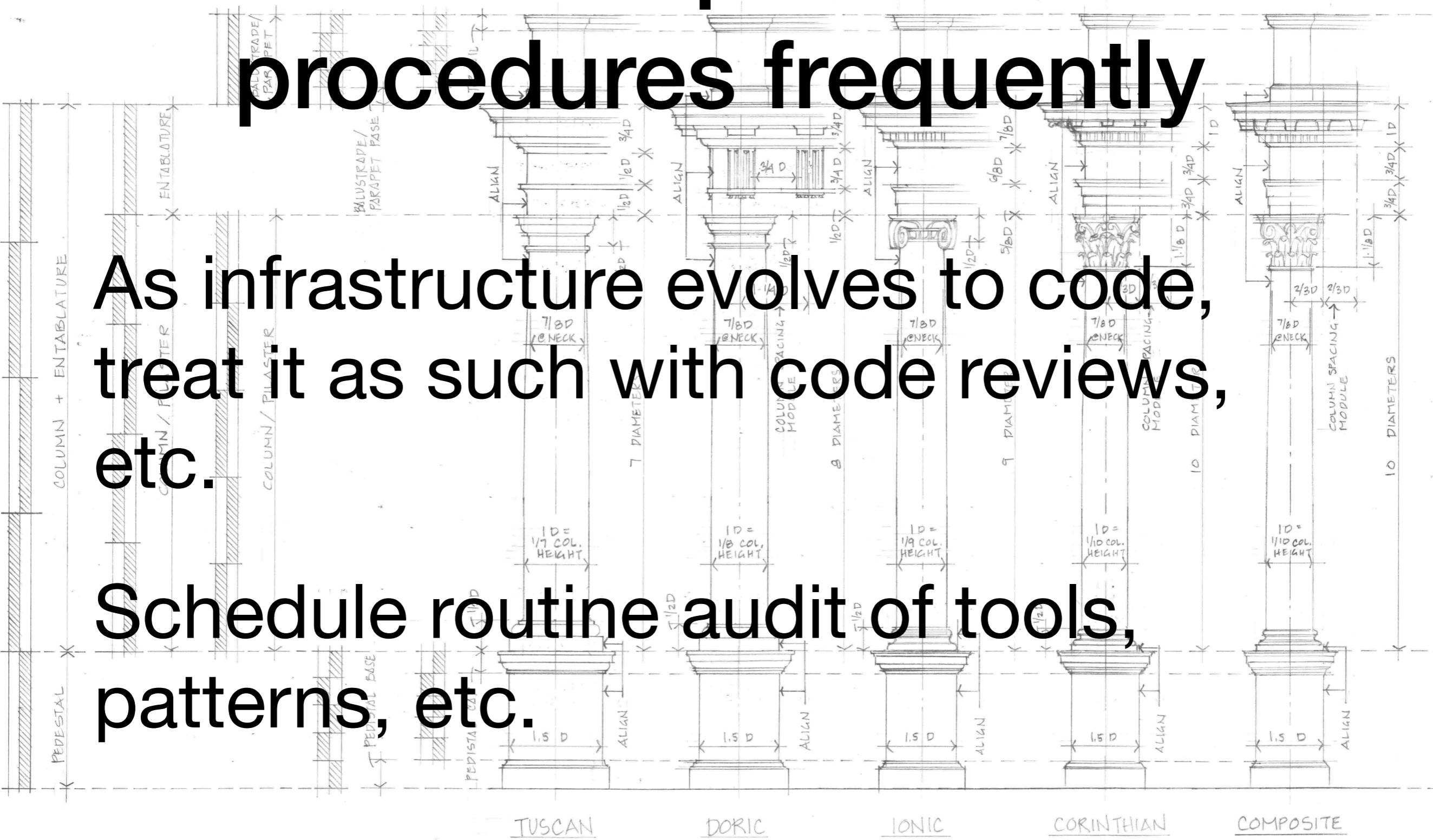
# 'and' is a commit smell



# Refine operations procedures frequently

As infrastructure evolves to code,  
treat it as such with code reviews,  
etc.

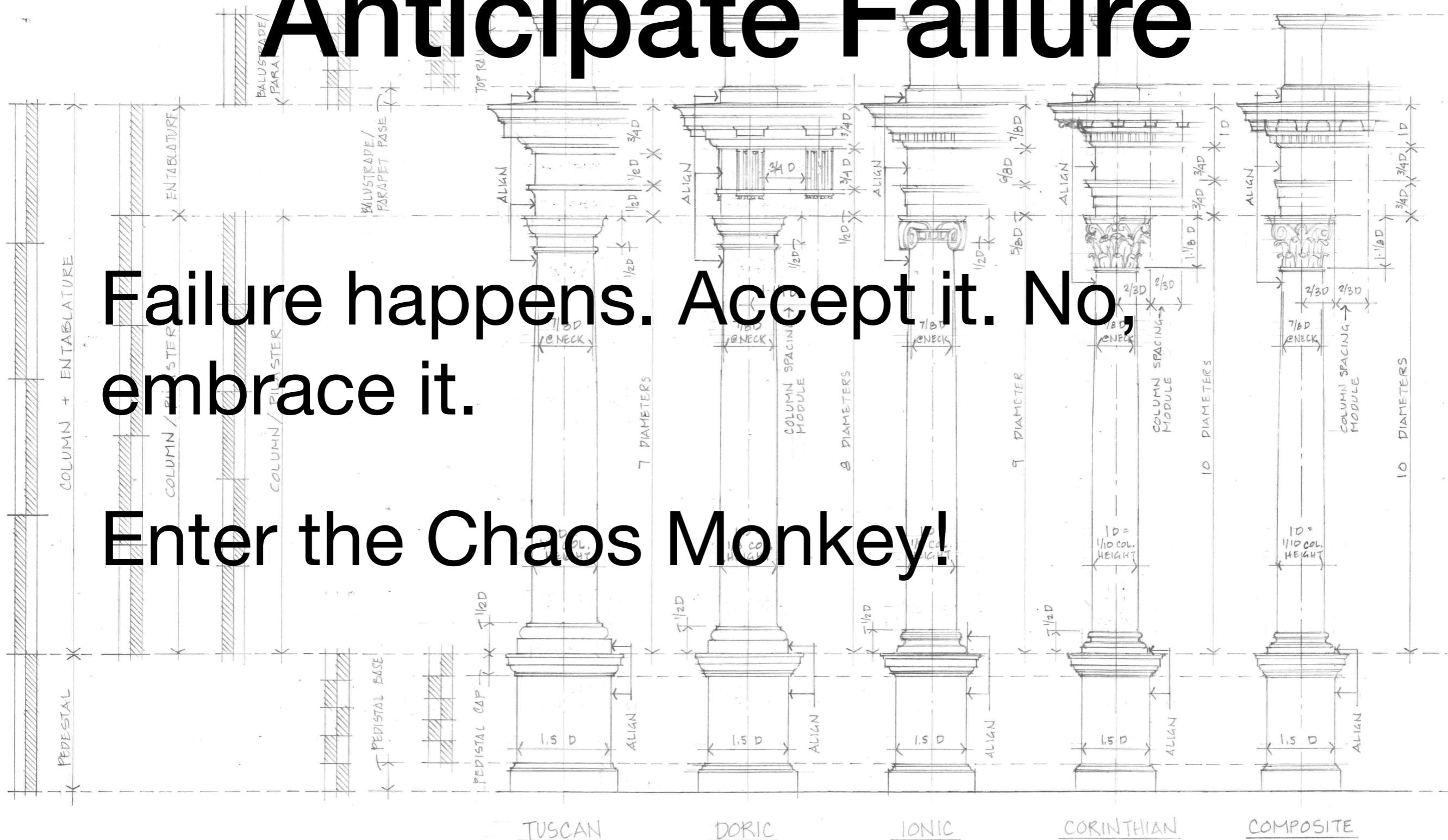
Schedule routine audit of tools,  
patterns, etc.



# Anticipate Failure

Failure happens. Accept it. No,  
embrace it.

# Enter the Chaos Monkey!

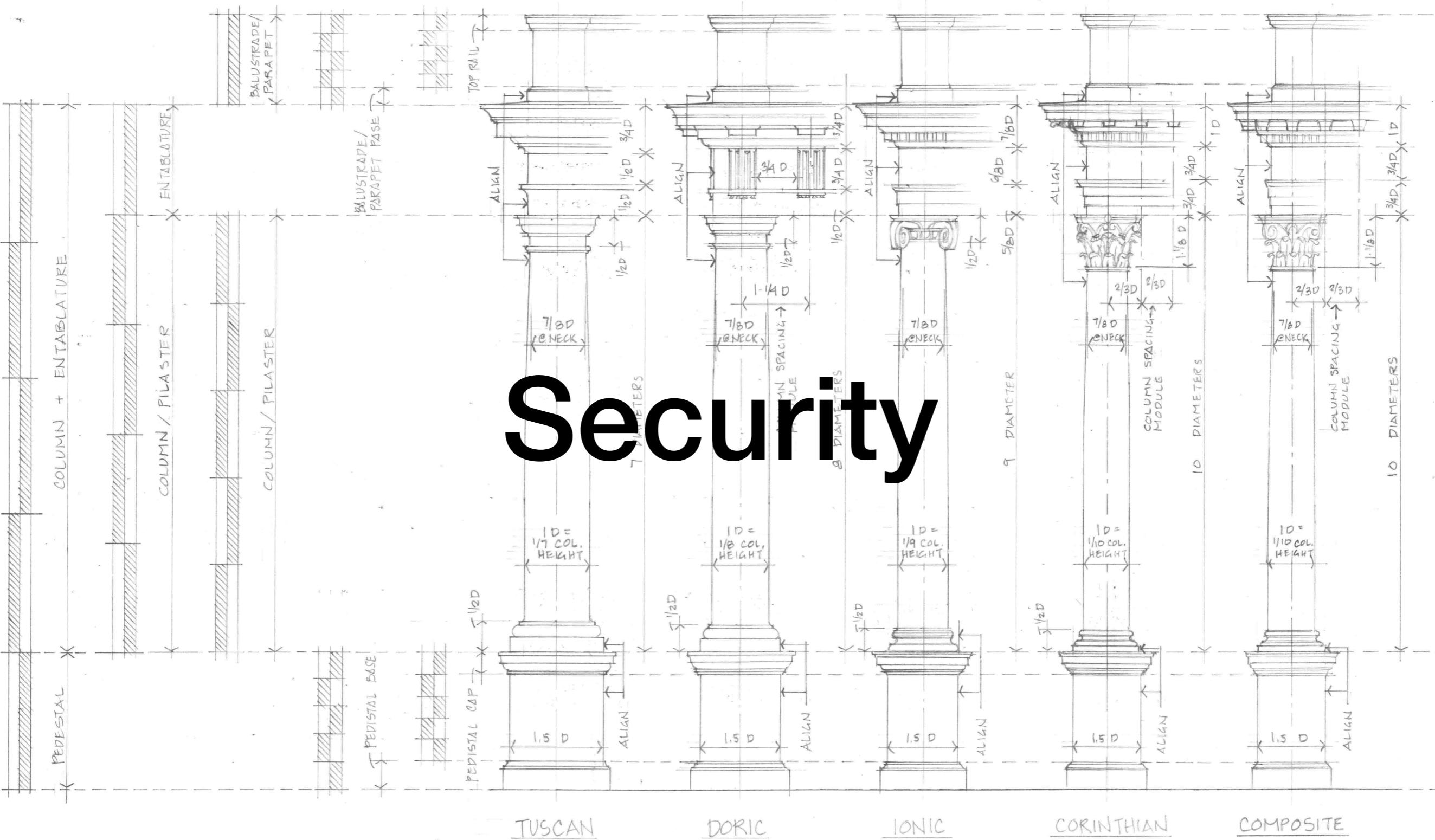


# Learn from all operational failures

It is only a mistake if you don't learn from it.



# Security

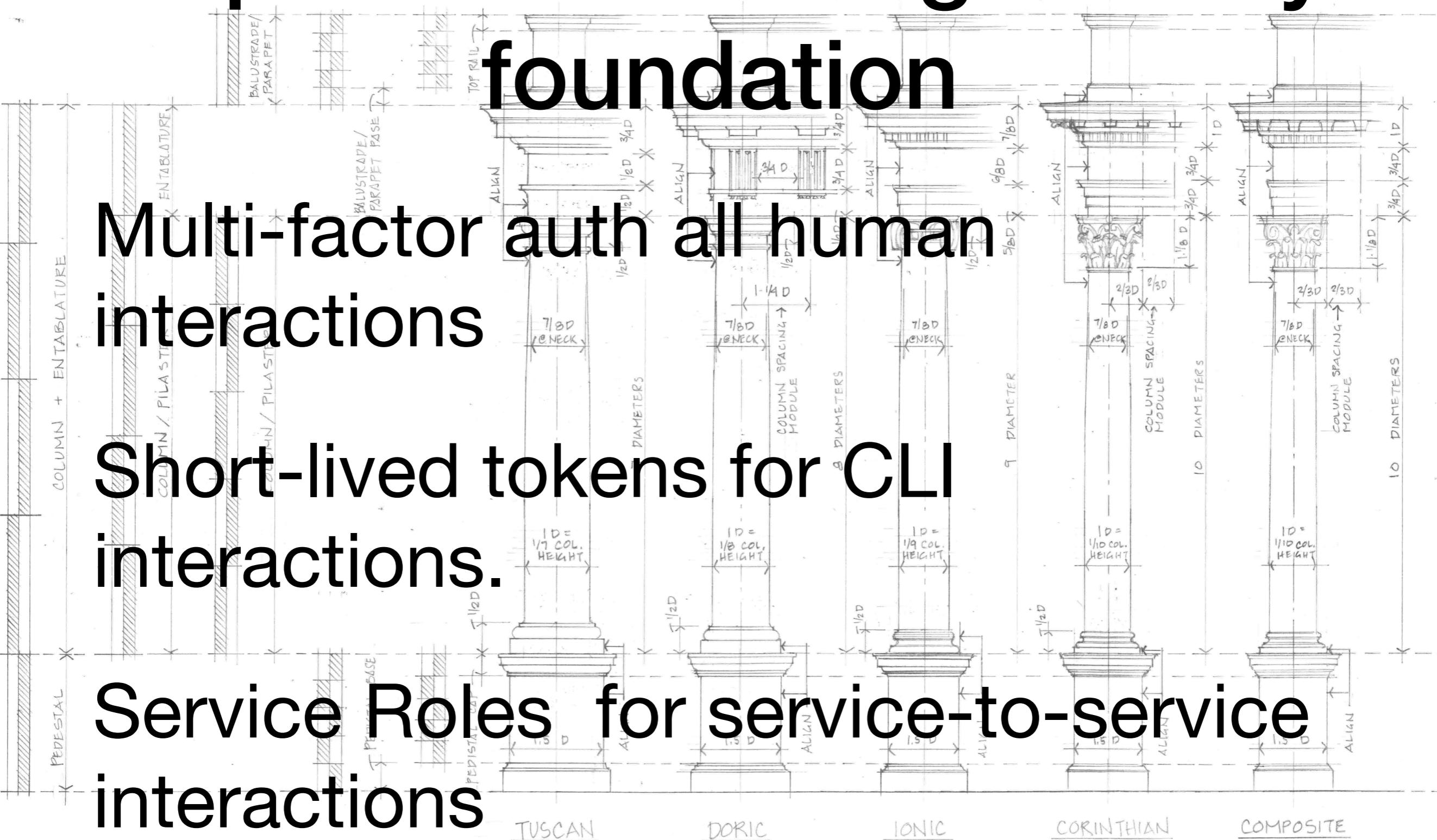


# Implement a strong identity foundation

# Multi-factor auth all human interactions

# Short-lived tokens for CLI interactions.

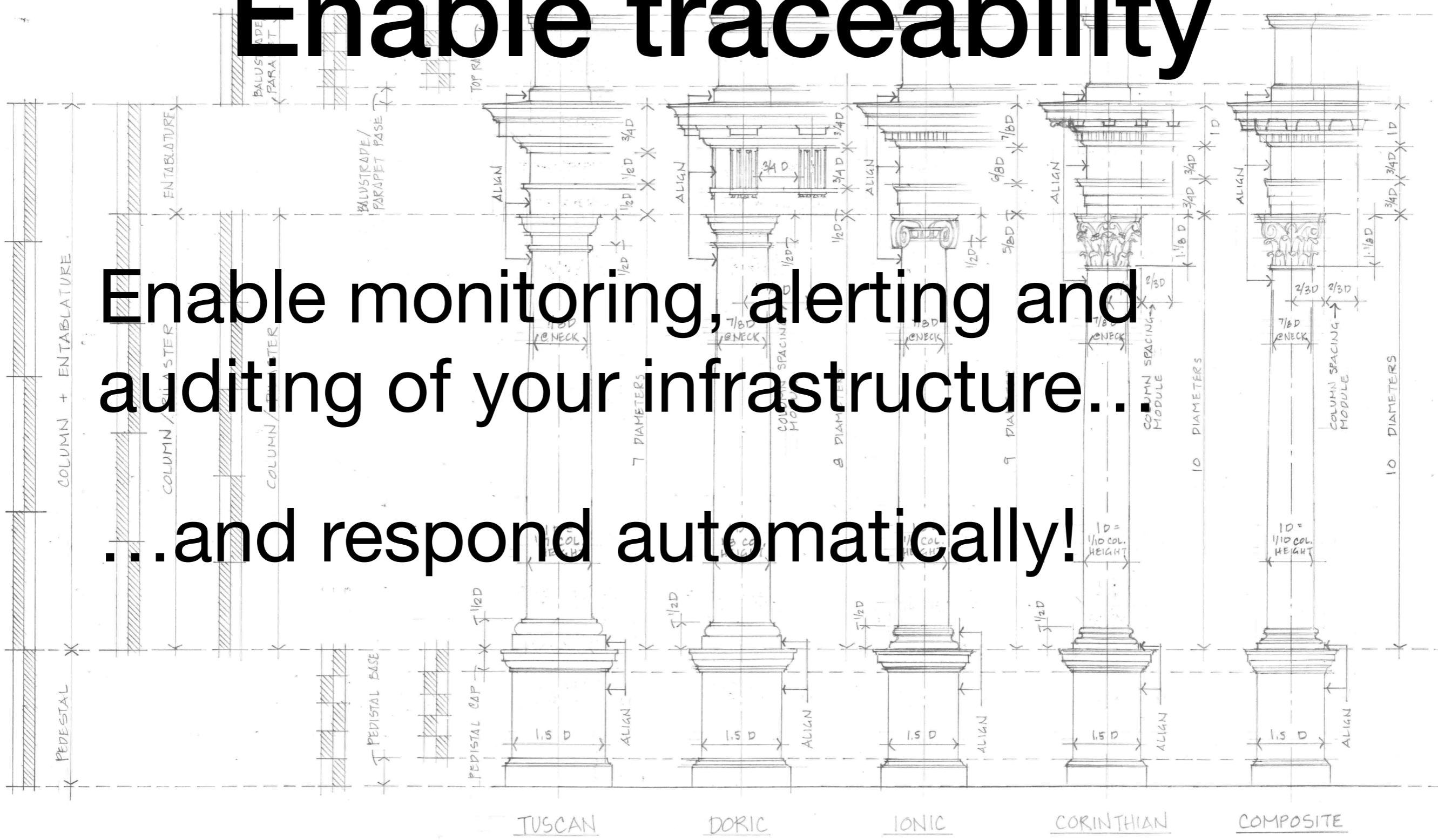
# Service Roles for service-to-service interactions



# Enable traceability

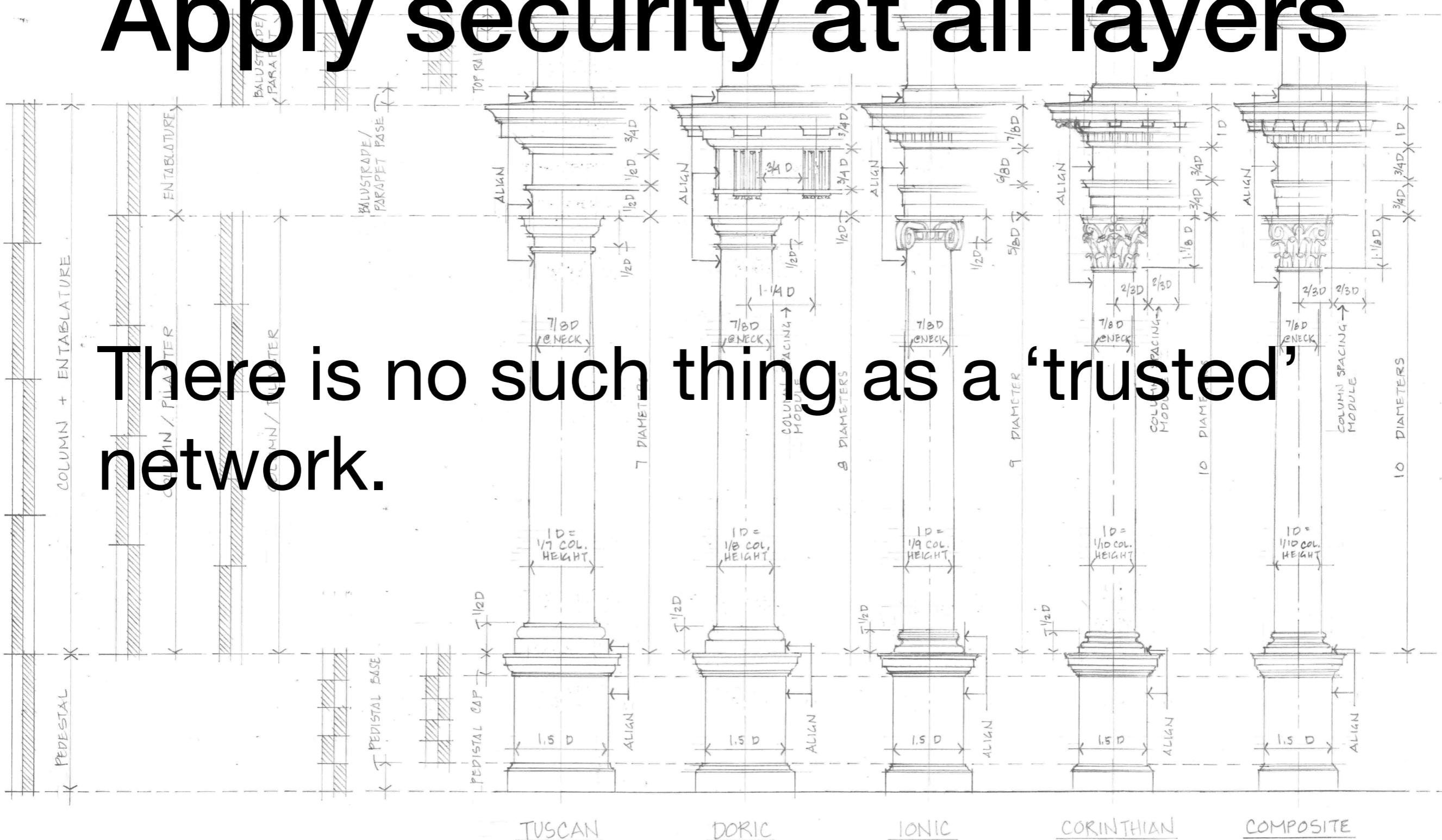
Enable monitoring, alerting and auditing of your infrastructure..

...and respond automatically!



# Apply security at all layers

There is no such thing as a ‘trusted’ network.



# Automate security best practices

Best Practices are constantly changing. (Apply the ones that are relevant to you, ignore the ones that don't)

Security controls as code.

TUSCAN

DORIC

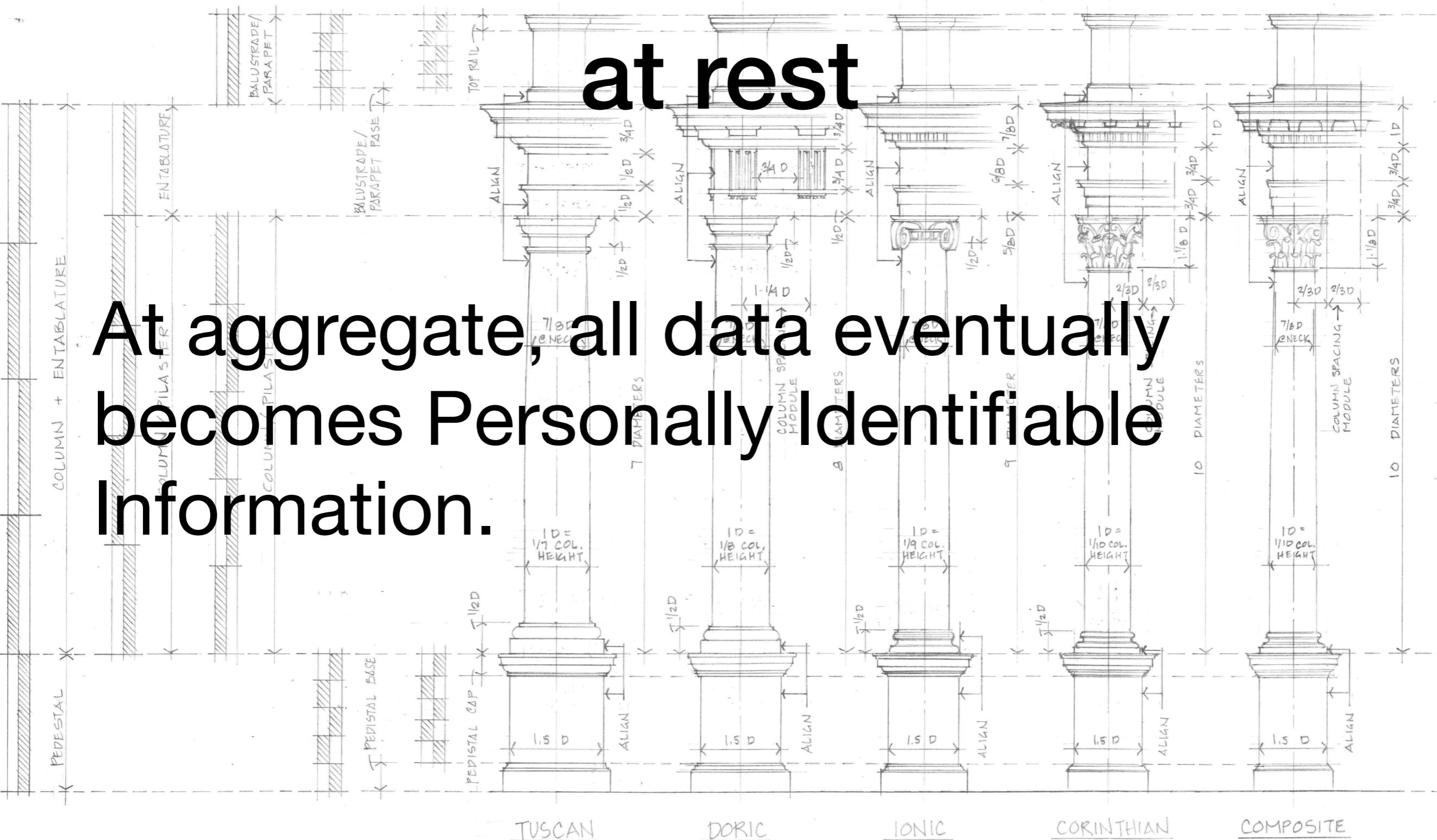
IONIC

CORINTHIAN

COMPOSITE

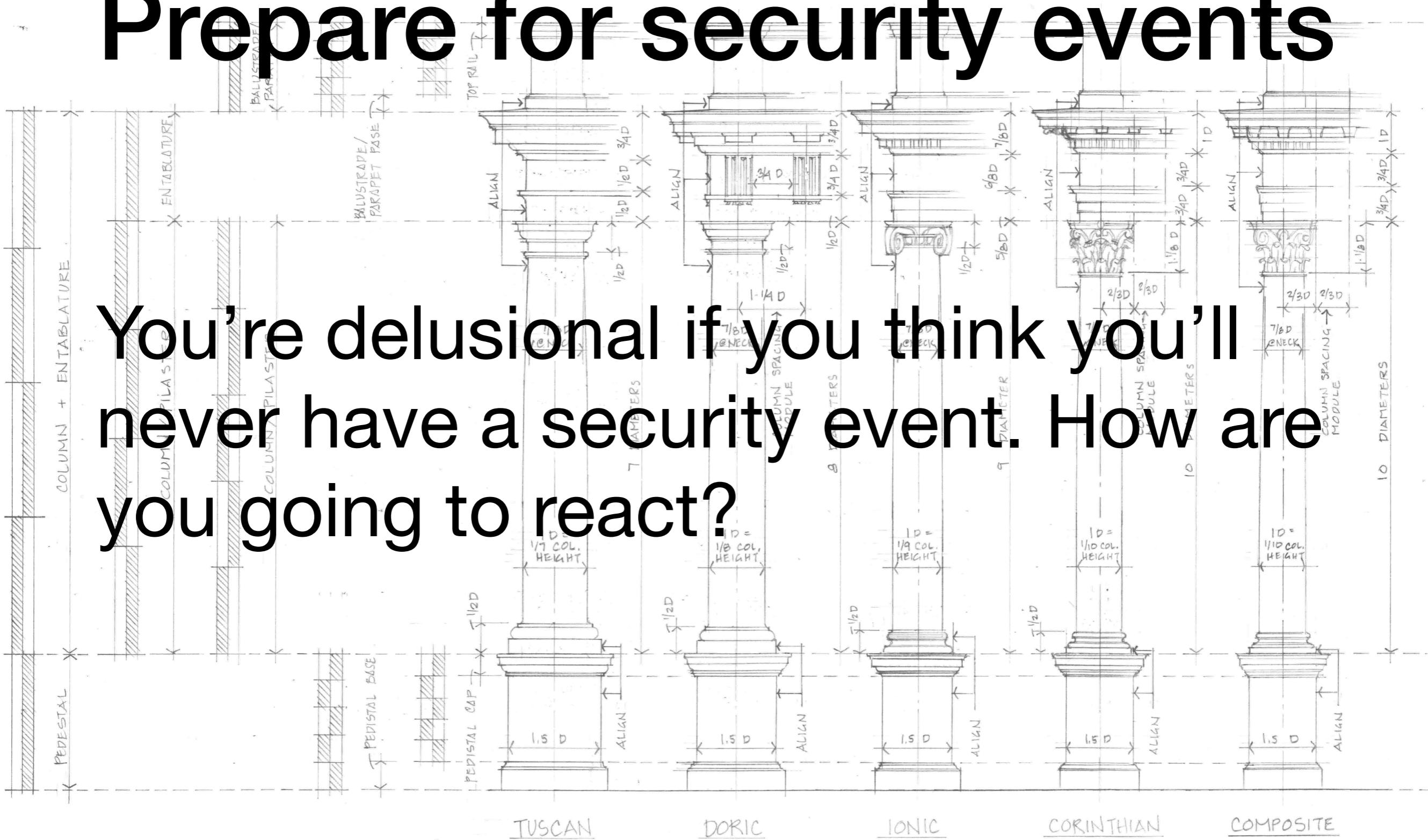
# Protect data in transit and at rest

At aggregate, all data eventually becomes Personally Identifiable Information.

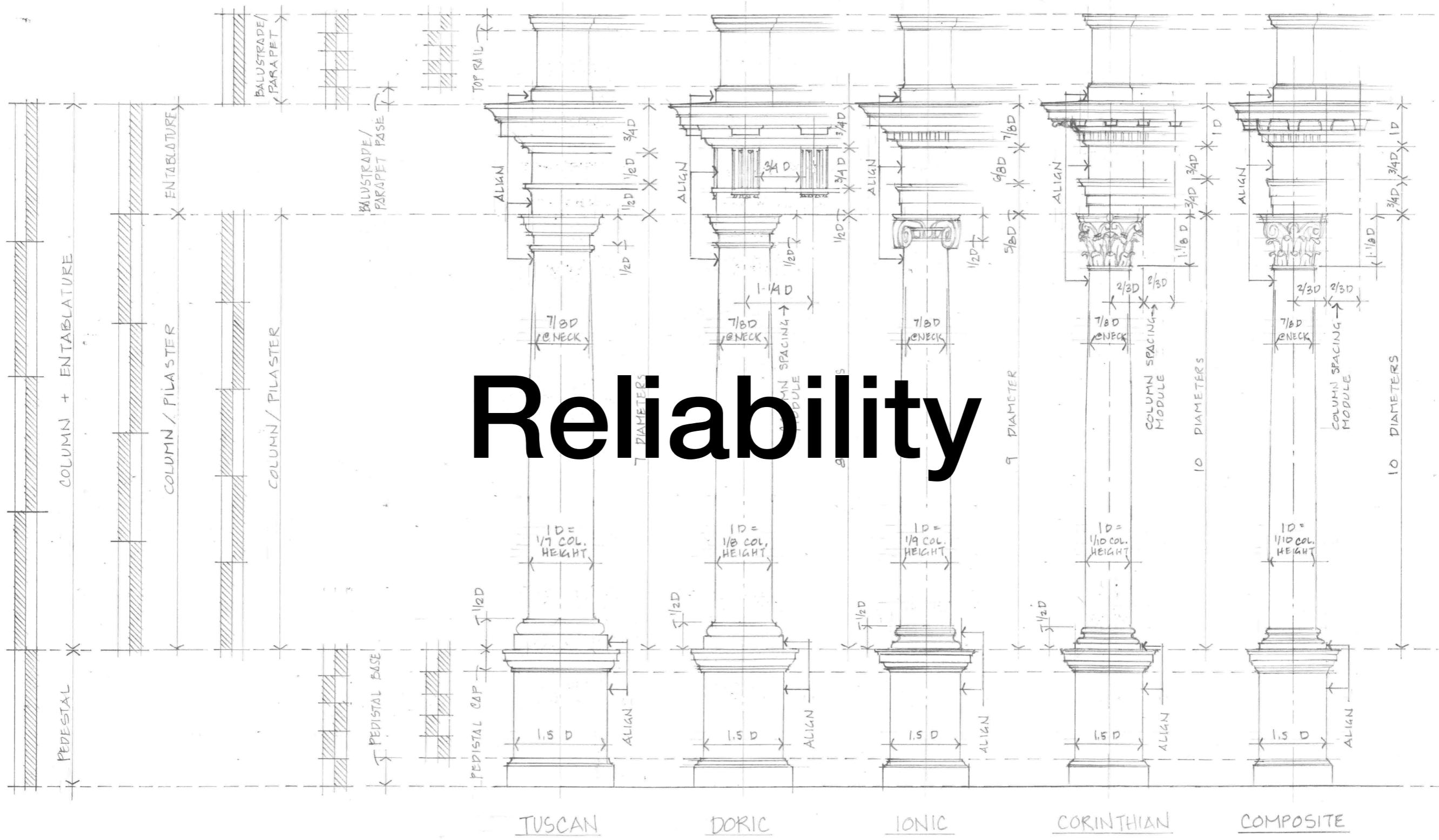


# Prepare for security events

You're delusional if you think you'll never have a security event. How are you going to react?

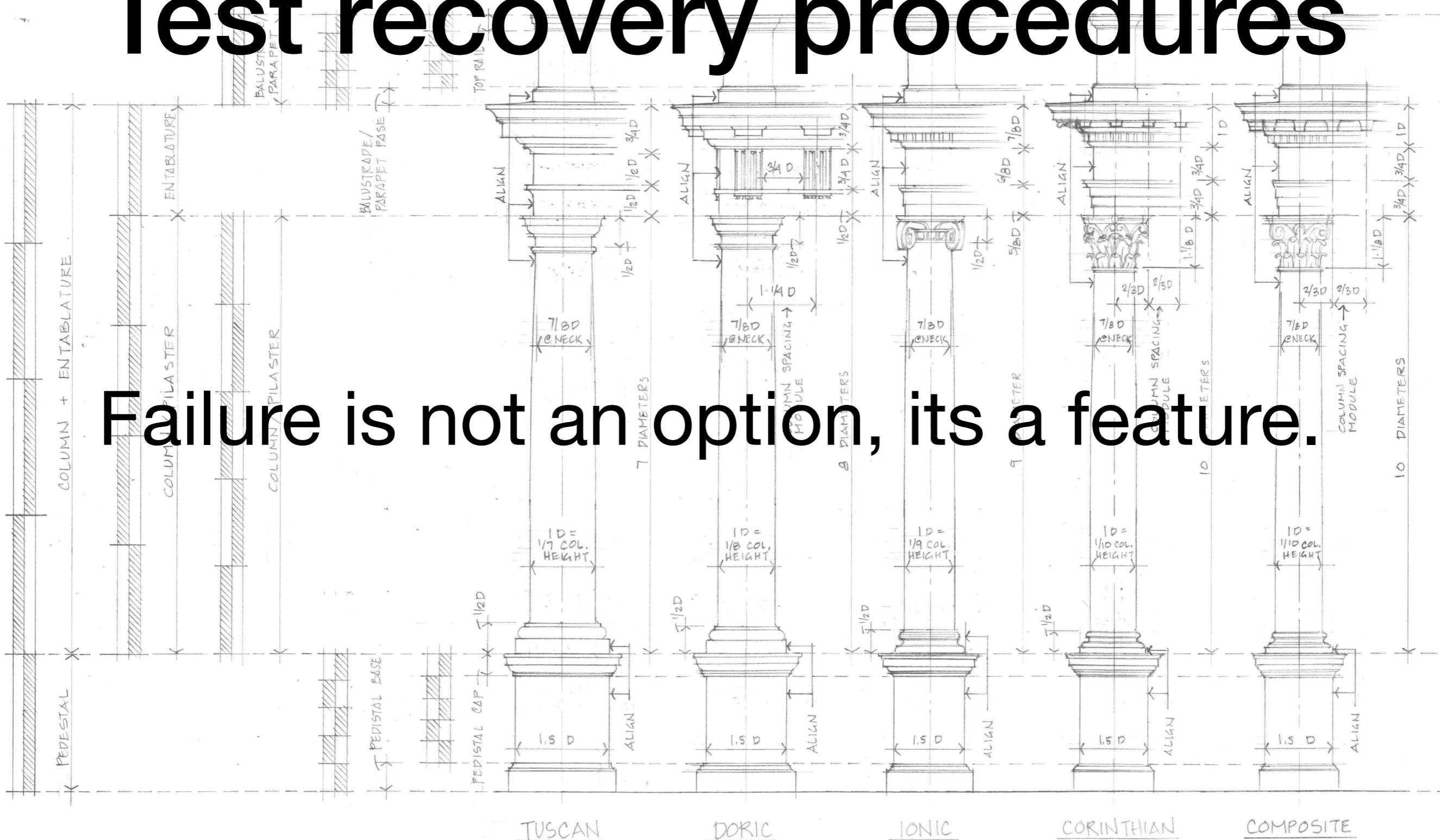


# Reliability



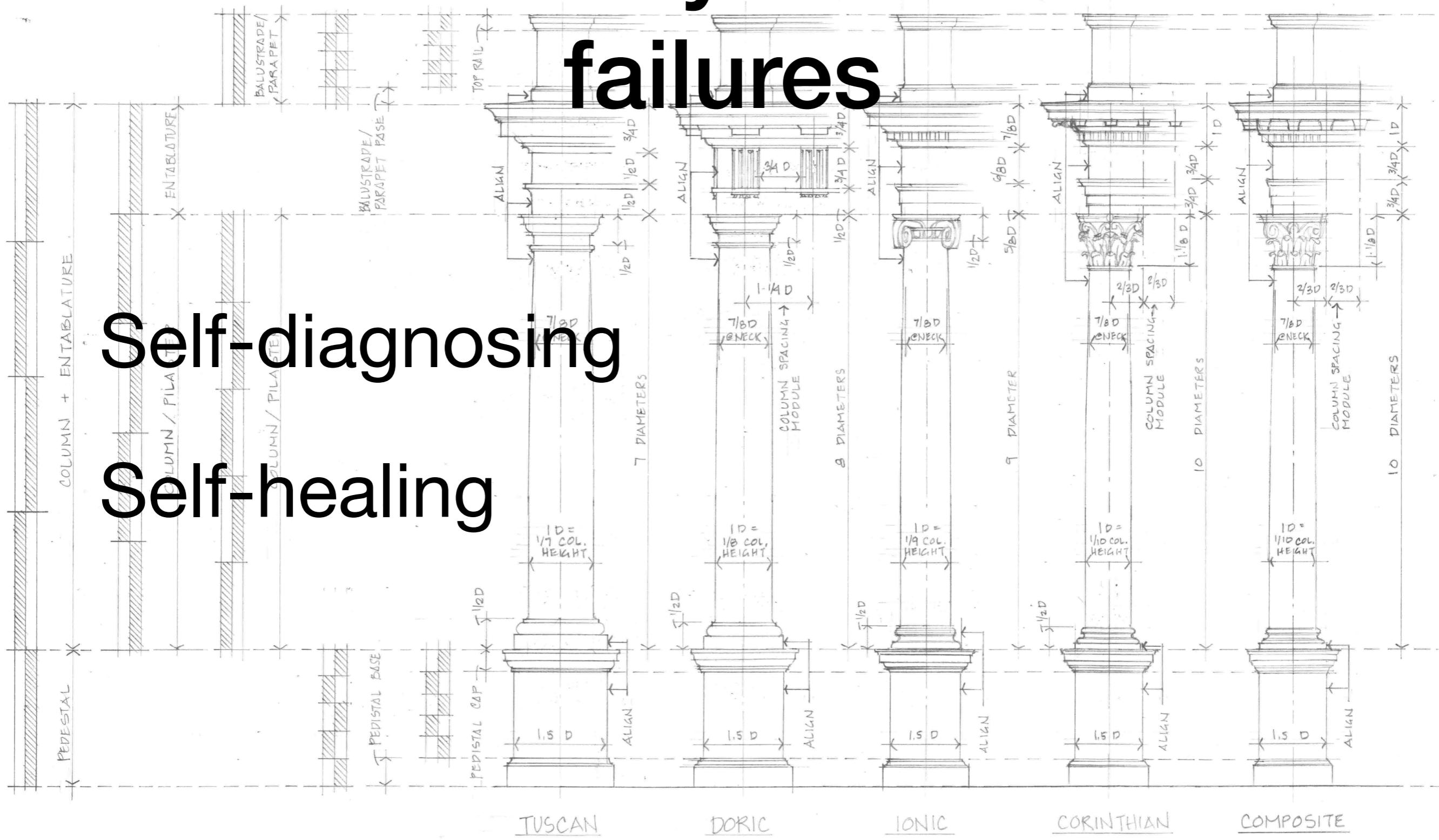
# Test recovery procedures

**Failure is not an option, its a feature.**



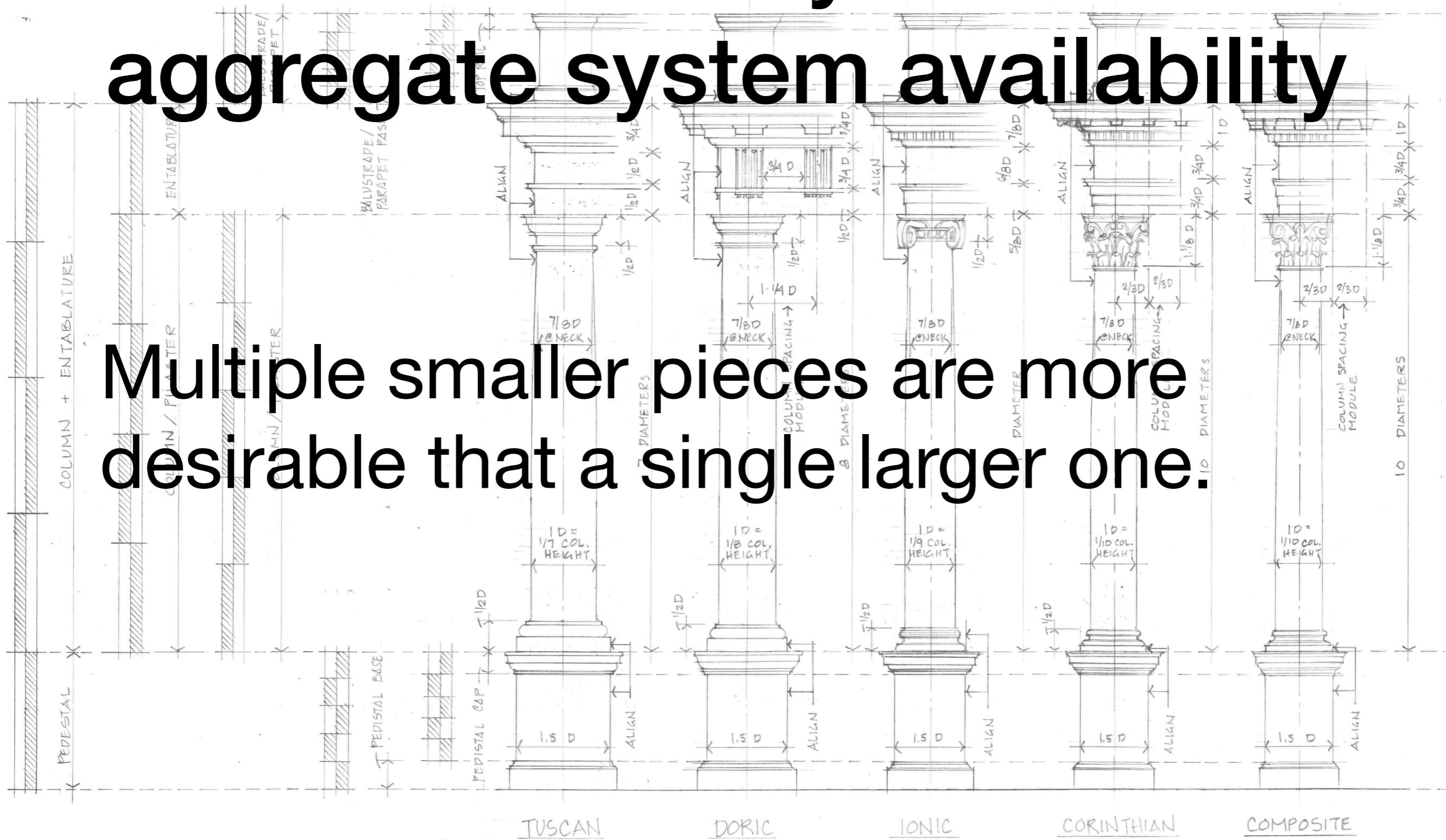
# Automatically recover from failures

# Self-diagnosing Self-healing



# Scale horizontally to increase aggregate system availability

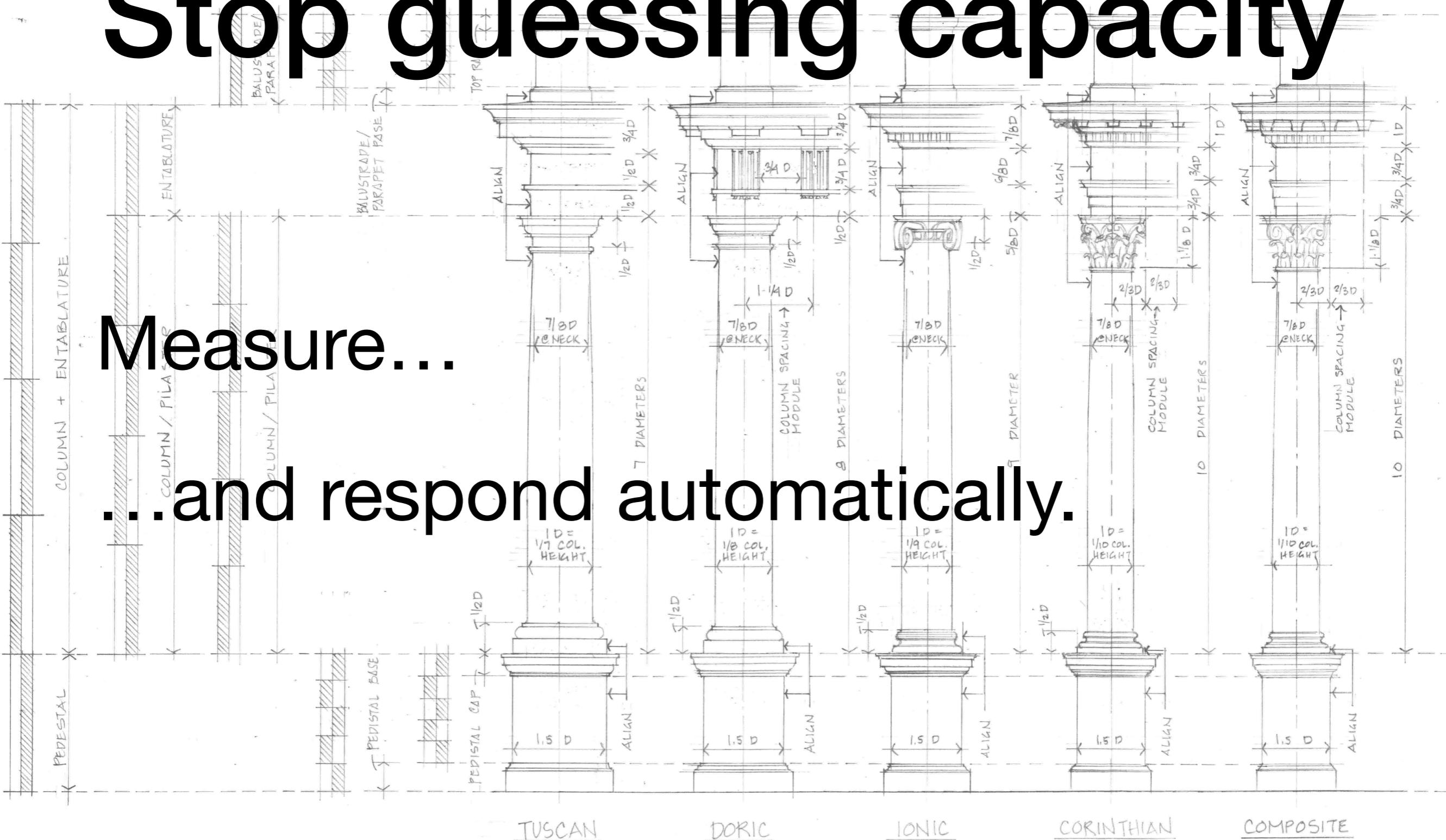
Multiple smaller pieces are more desirable than a single larger one.



# Stop guessing capacity

Measure...

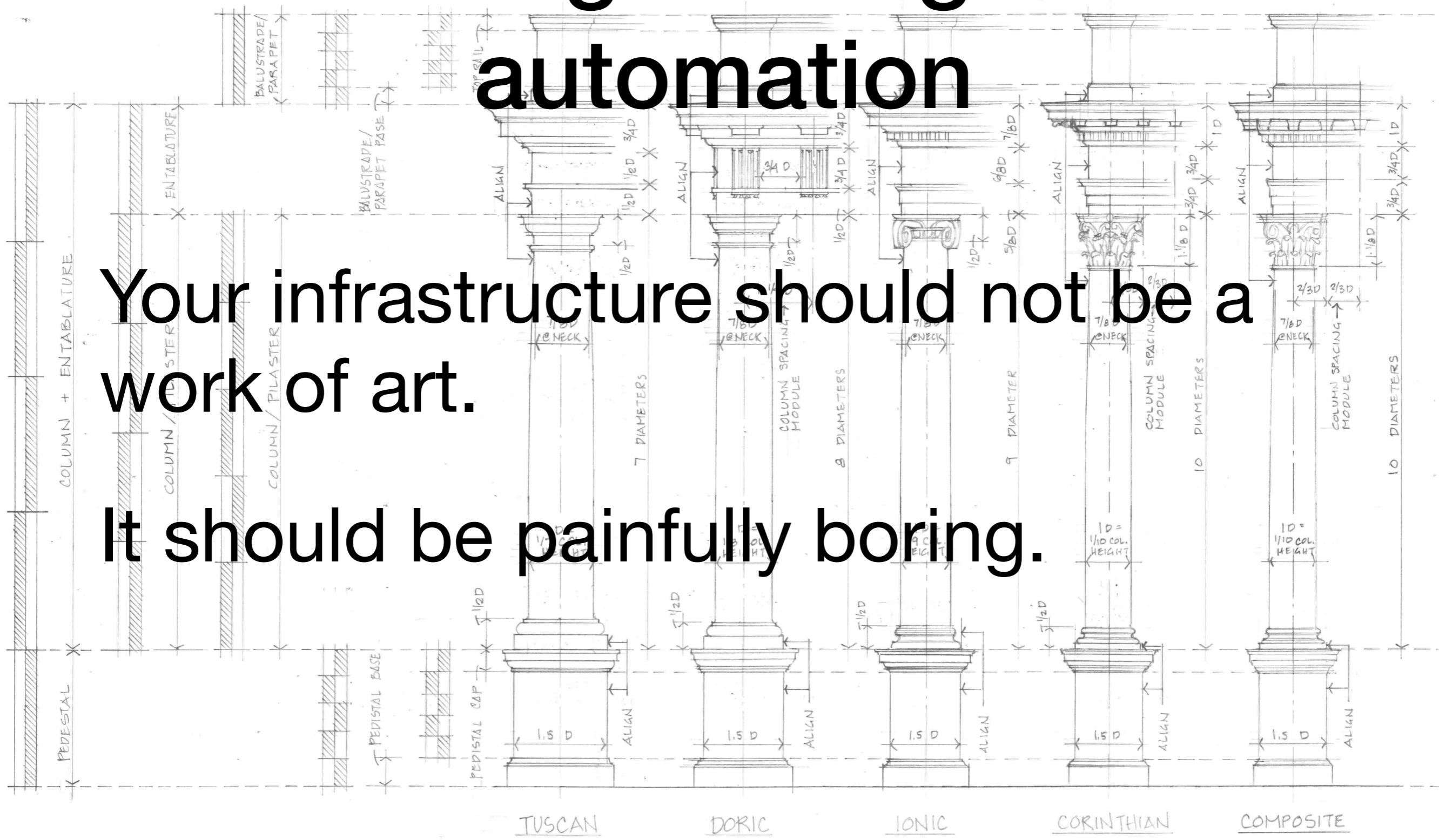
...and respond automatically.



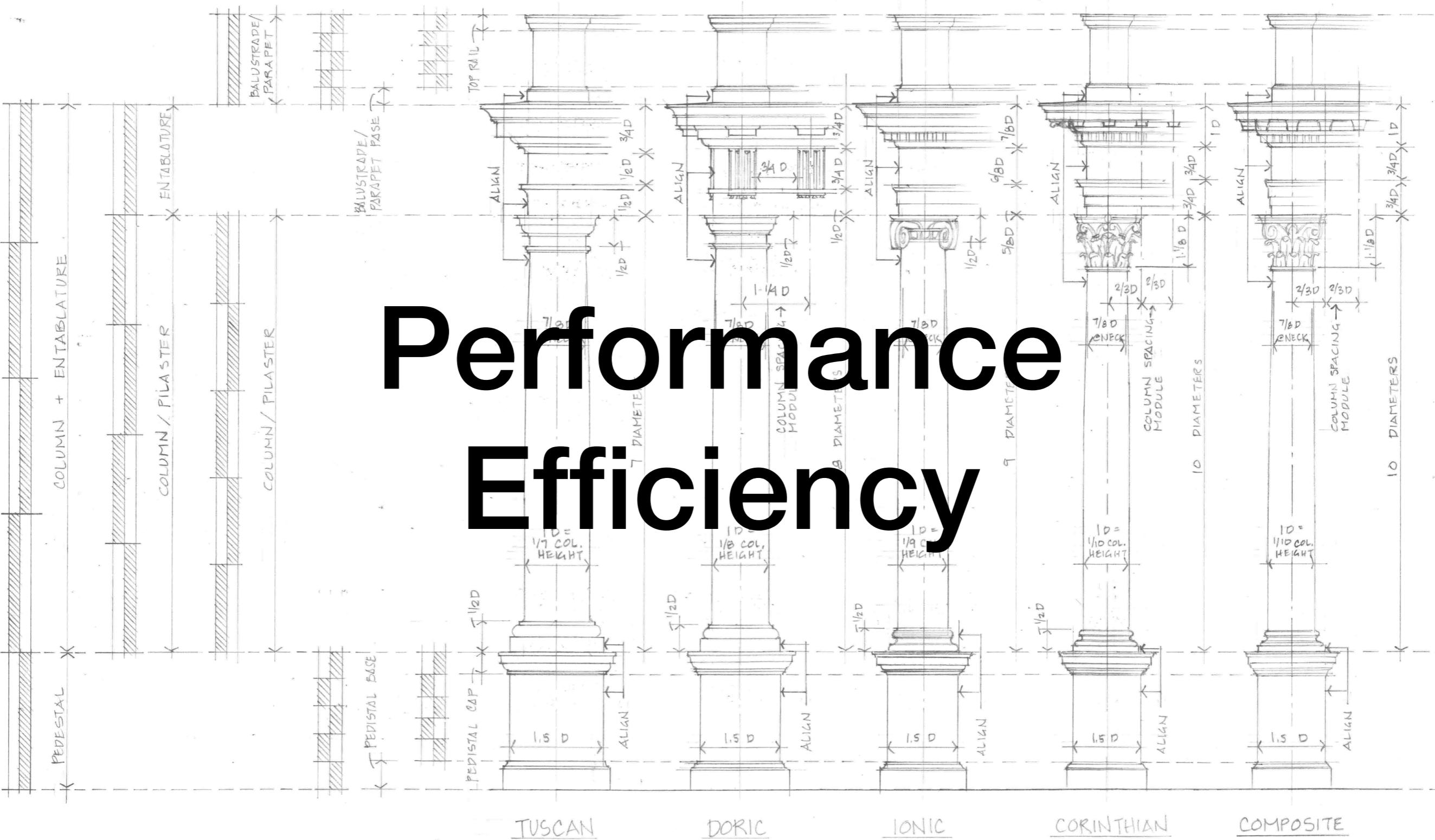
# Manage change in automation

# Your infrastructure should not be a work of art.

It should be painfully boring.



# Performance Efficiency



# Democratize advanced technologies

As cool as learning a new tech is,  
your automation team likely doesn't  
need to understand how it works.

(Beyond how to test with it.)

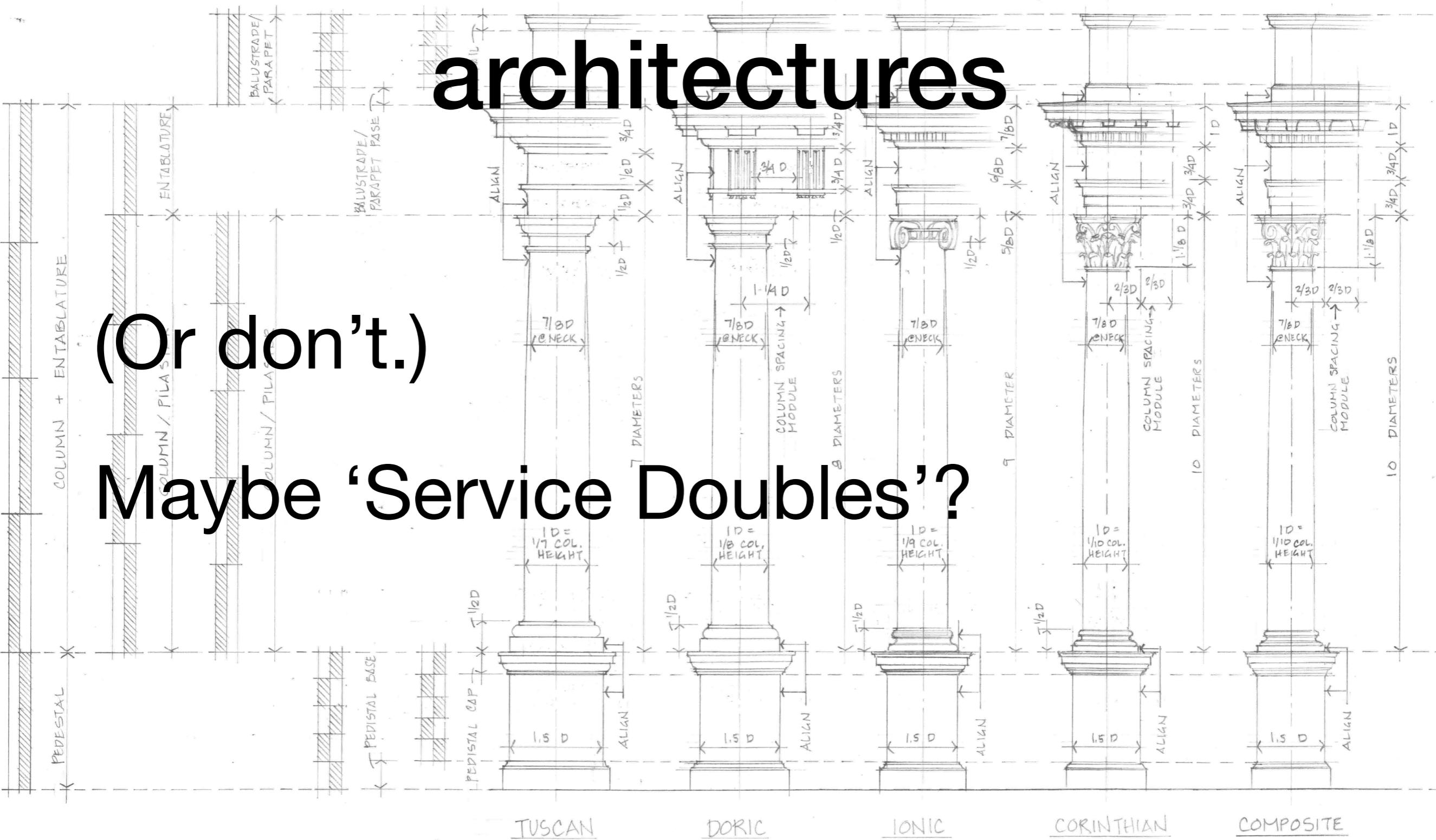
# Go global in minutes

If your production infrastructure is globally distributed, so should your automation infrastructure.

# Use serverless architectures

(Or don't.)

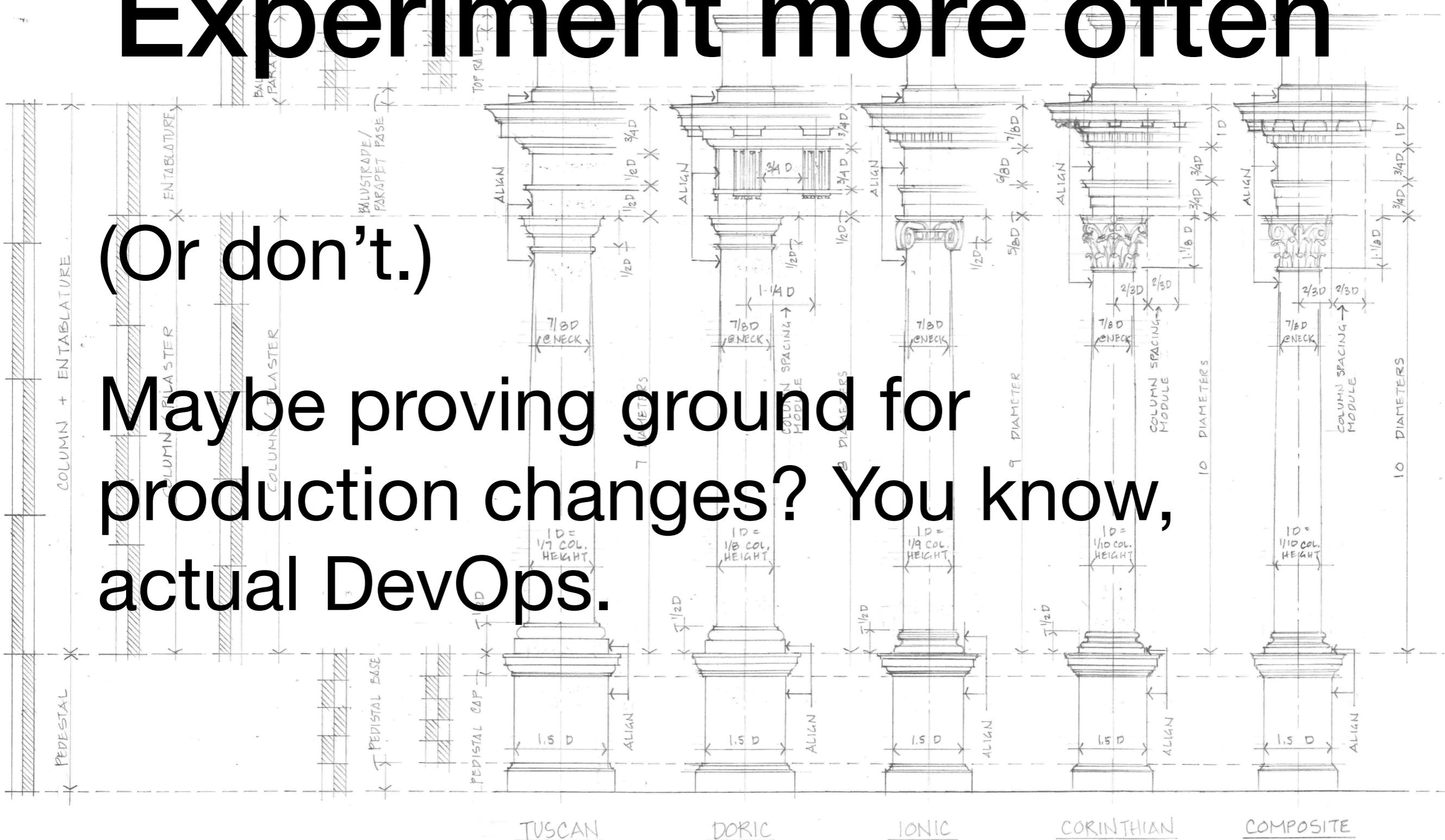
# Maybe ‘Service Doubles’?



# Experiment more often

(Or don't.)

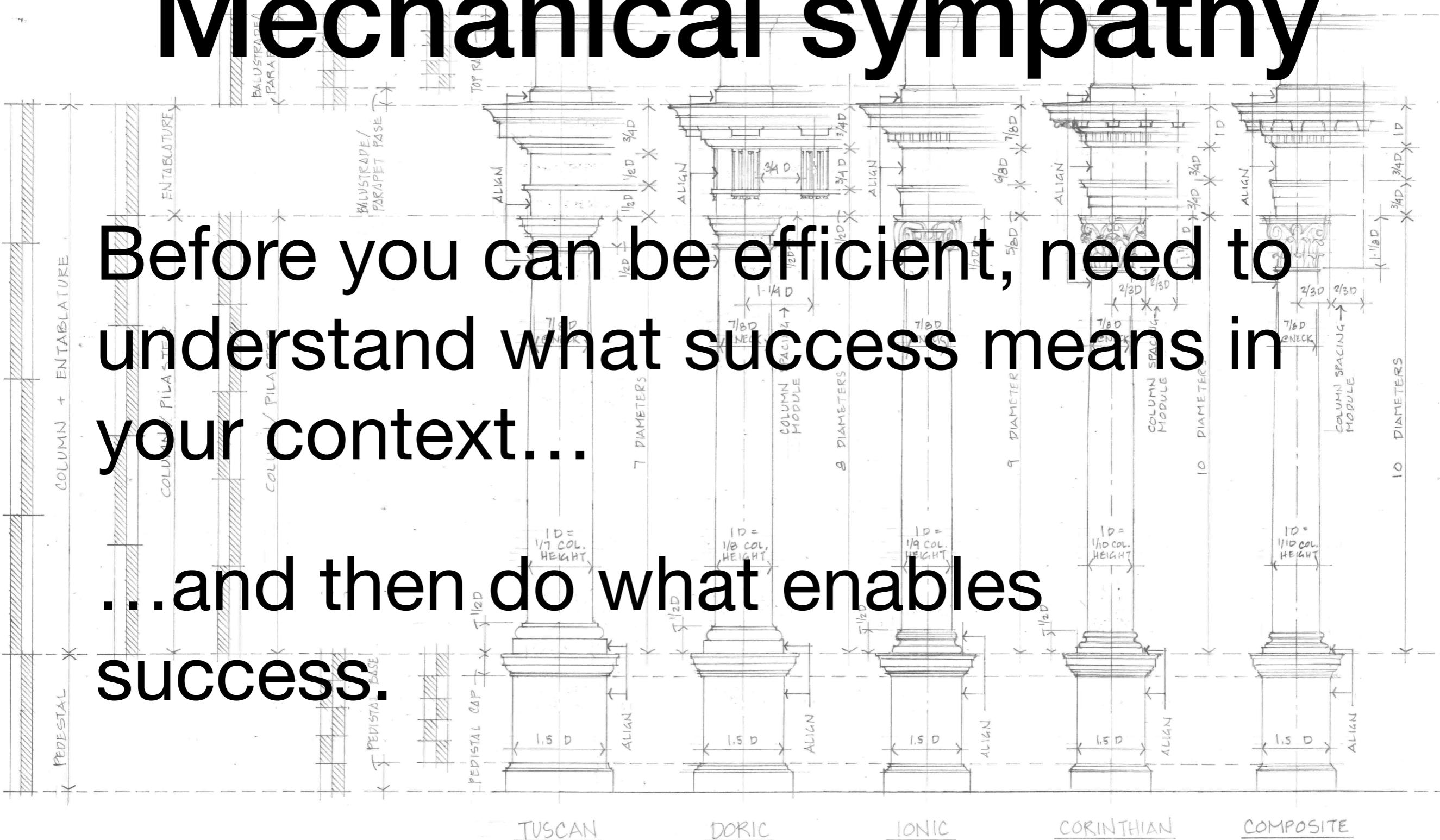
Maybe proving ground for production changes? You know, actual DevOps.



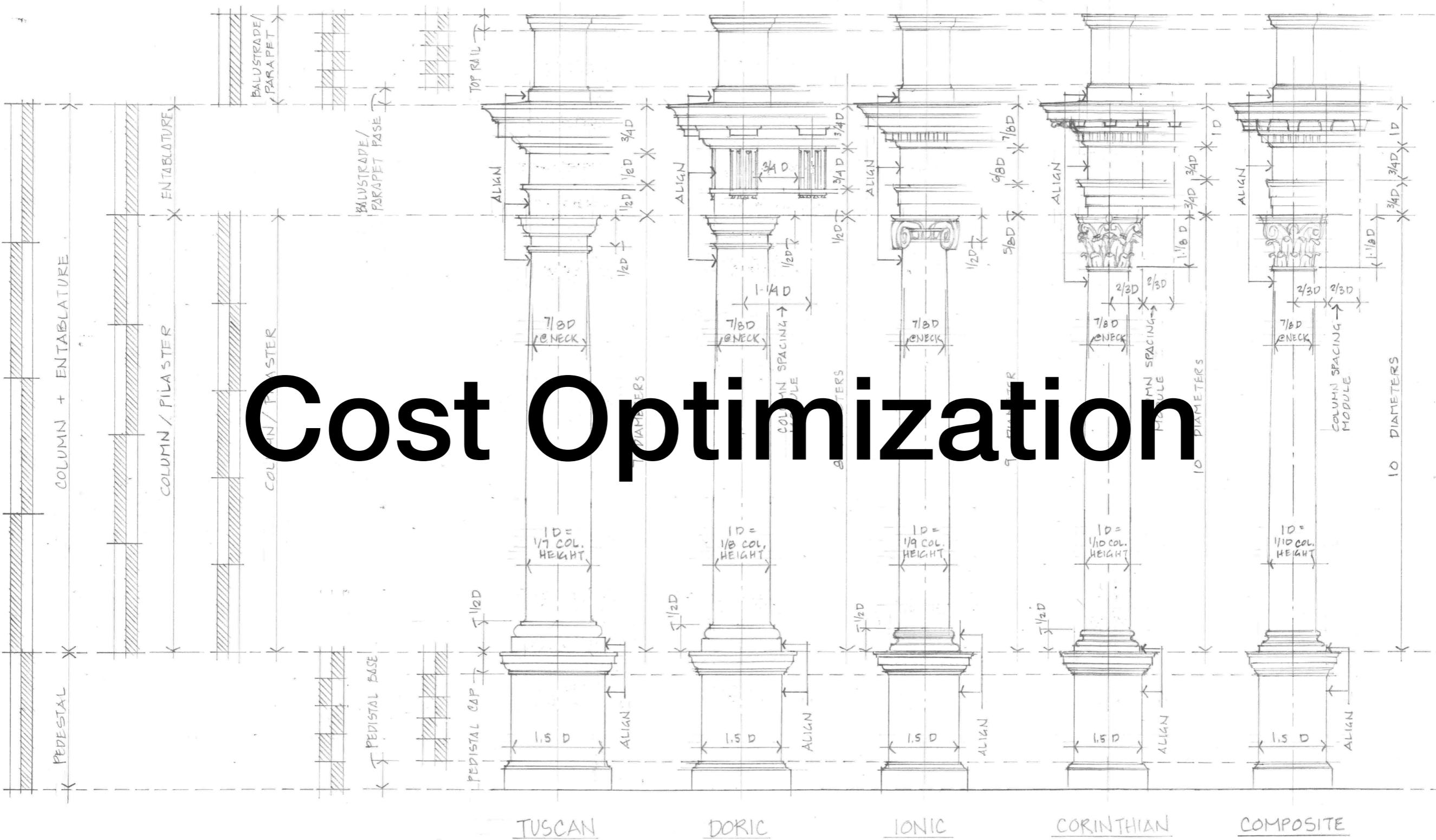
# Mechanical sympathy

Before you can be efficient, need to understand what success means in your context...

...and then do what enables success.

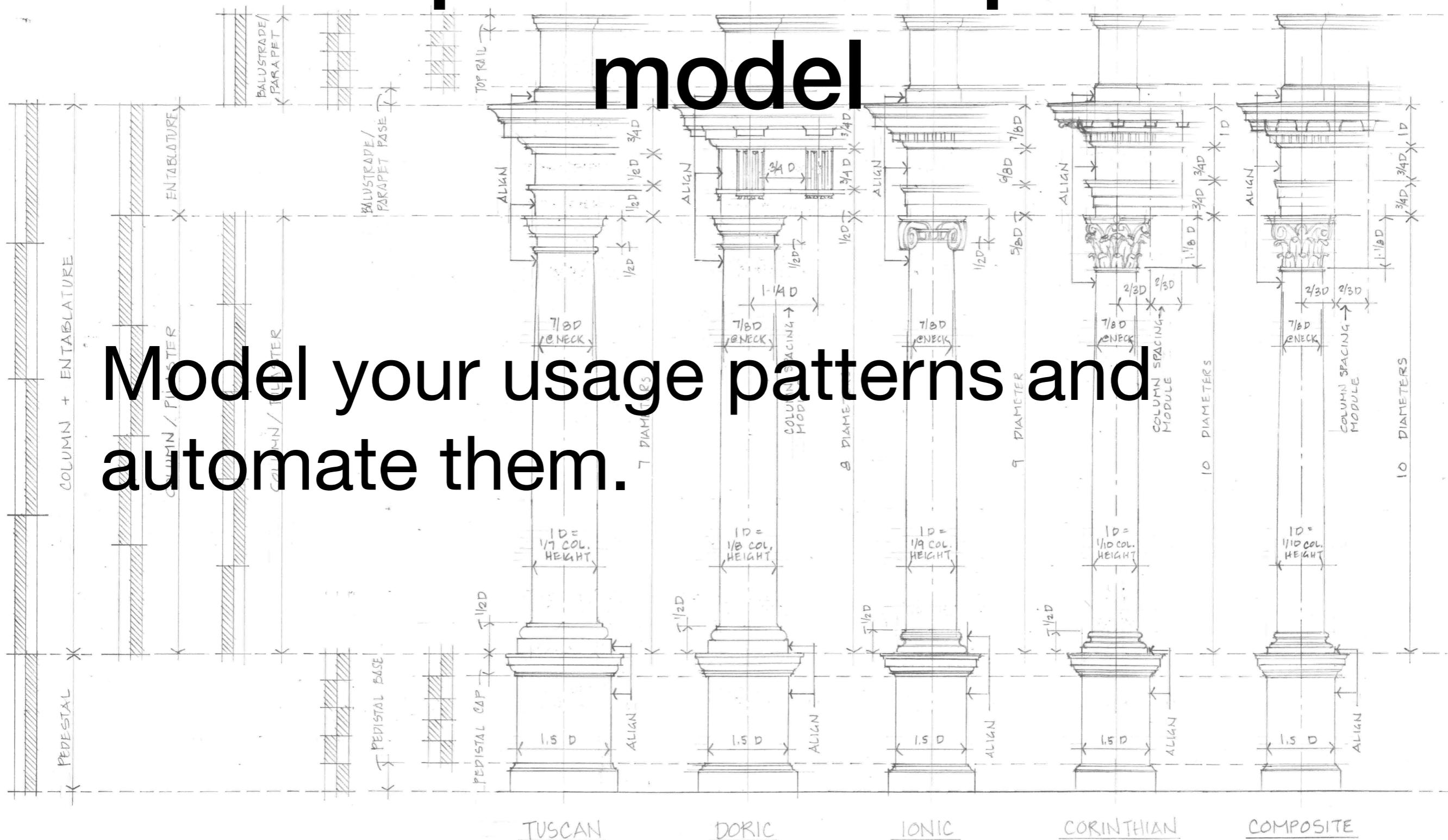


# Cost Optimization



# Adopt a consumption model

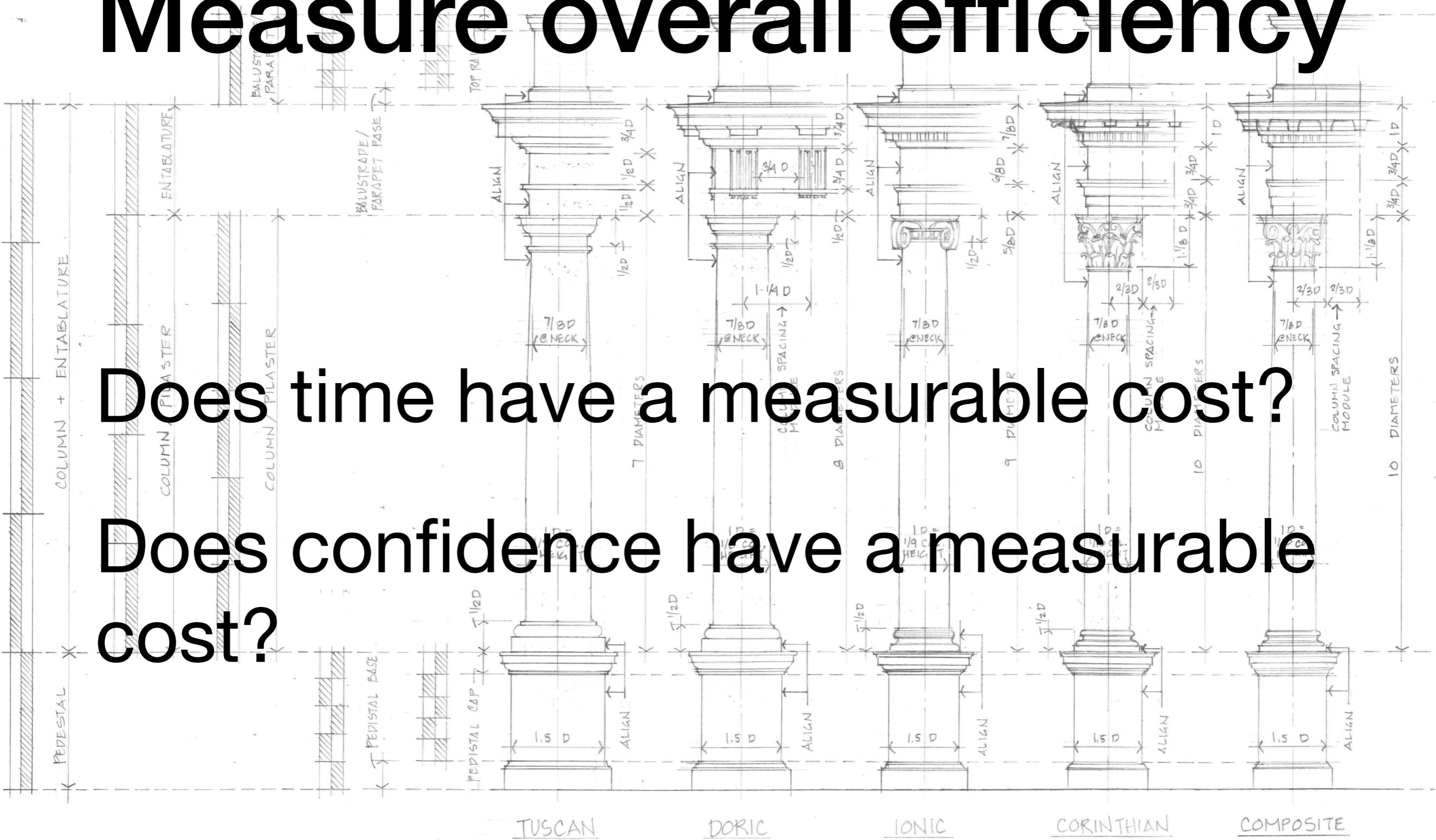
# Model your usage patterns and automate them.



# Measure overall efficiency

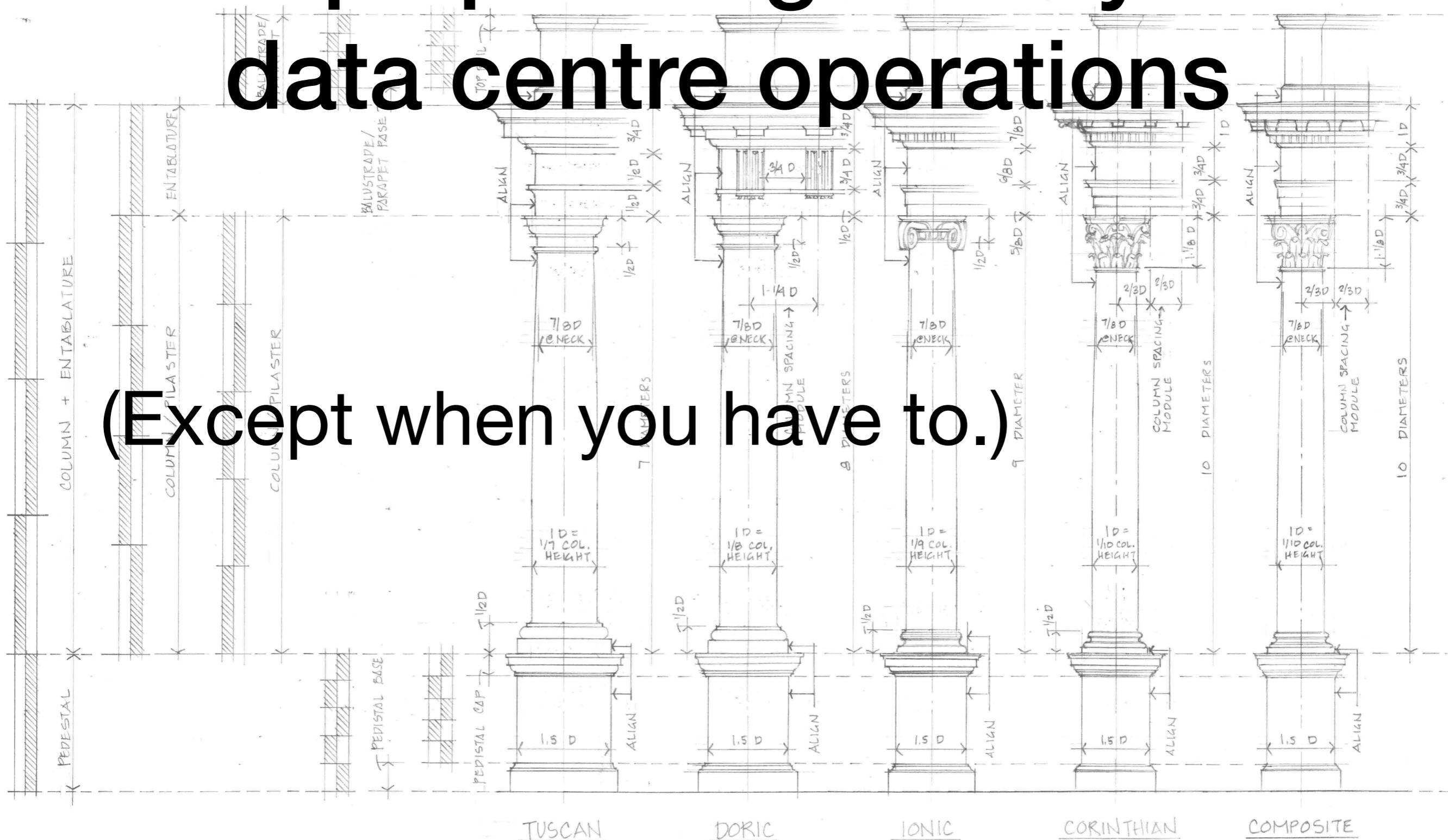
Does time have a measurable cost?

Does confidence have a measurable cost?



# Stop spending money on data centre operations

(Except when you have to.)



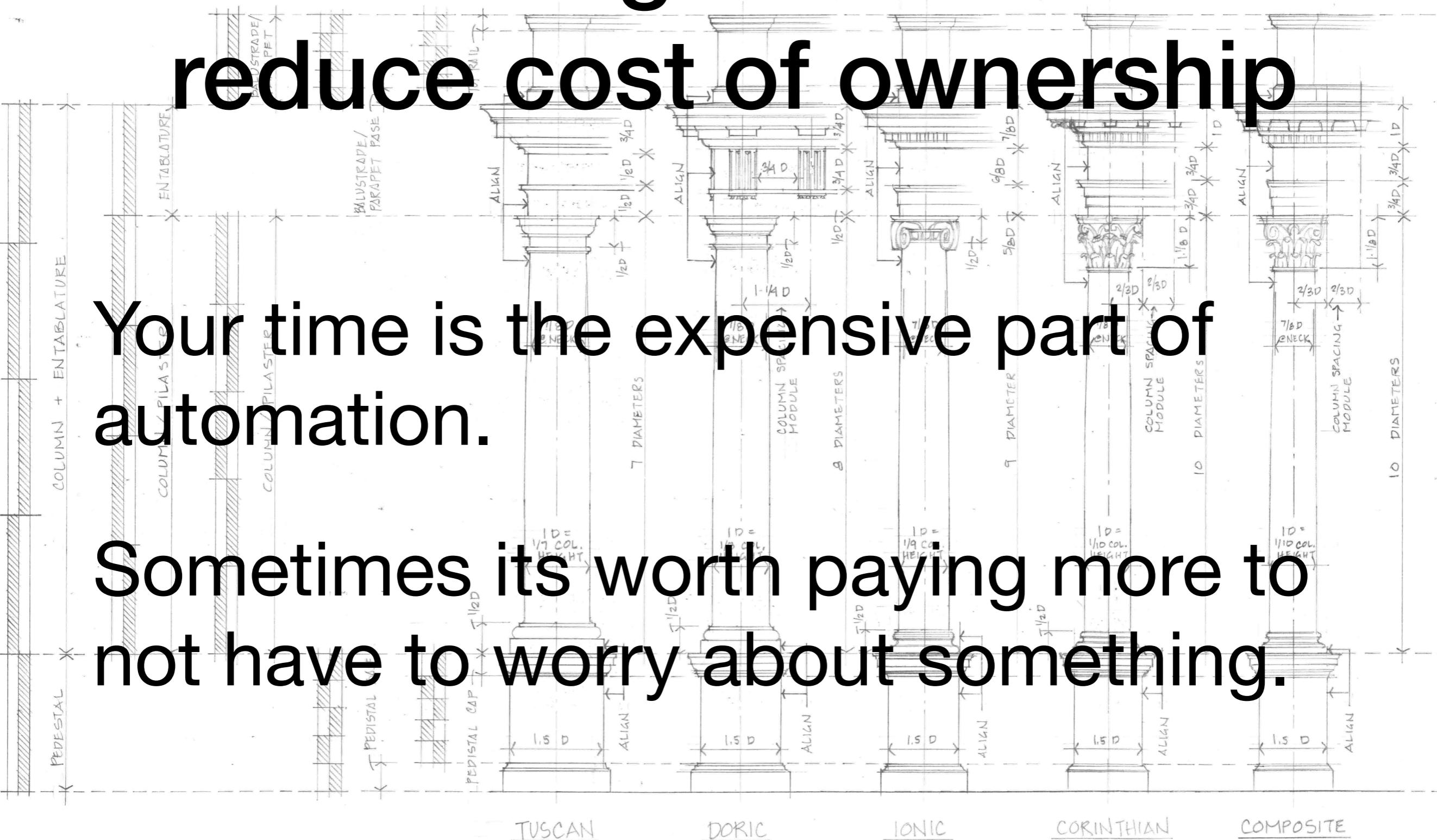
# Analyze and attribute expenditure

# 'How much does our automation infrastructure cost' should be an answerable question.

# Use managed services to reduce cost of ownership

Your time is the expensive part of automation.

Sometimes its worth paying more to not have to worry about something.



# So. Are you We Architected?

