

A large red square with a white border. Inside the square, the text 'W4156' is centered in a large, white, sans-serif font. Below it, the text 'Styles, Patterns and Reference Architectures' is centered in a smaller, white, serif font.

W4156

**Styles, Patterns and
Reference Architectures**

Architecture Recap

<p>“The Architecture” (shared understanding / artifact)</p>	<p>The “important” set of decisions throughout a software project that enable</p> <ol style="list-style-type: none">1. Ensure/derisk the project/iteration is a success / meets key Q.A2. Captured in some artifact which effectively communicates to stakeholders <p>[1. Architecture, Architecture vs Design & Architecturally Significant 2. Views]</p>
<p>“Architecting” (the process)</p>	<p>The <i>process</i> of taking a project from idea/requirements through to architecture, construction and managing evolution of the architecture in response to evolving requirements</p> <p>(Requirements, [Quality Attributes], [Architecturally Significant Requirements], [Tradeoffs & Decisions], [Styles & Patterns]*, [Evolutionary Architecture], [Galls Law])</p>
<p>“Architect” (the responsibility)</p>	<p>Community organizer / engineering leader who leads this process, elevates the maturity of the team and may be responsible for trade-offs including challenging/balancing requirements.</p>

[] everything in brackets is a key topic we will cover

* Our dear friends patterns re-appear at a higher level of abstraction for system elements and how they can relate to each other. This can be considered an *input* to the architecture process

Agenda

Do we need to design each architecture from scratch?

In a similar way to ‘design patterns’ are there ‘catalogues’ of architectures with properties (pros and cons).


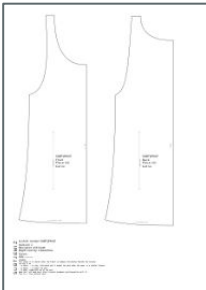

- ❑ Styles
- ❑ Patterns
- ❑ Reference Architectures

By the end of today I want you to understand key styles and patterns, be able to recognize them and understand the pros and cons of each

“Architecture Styles & Patterns”

Fashion

More Specific (and specialized)

	Fashion Style	'Flares Pattern'	My Blue Tuxedo Suit
Example		 wiki	
What is it?	"a distinctive appearance, typically determined by the principles according to which something is designed" [oxford dictionary]	"Bell-bottoms (or flares) are a style of trousers that become wider from the knees downward, forming a bell-like shape of the trouser leg" [wiki]	For a set of social occasions a specific item with cut, color, fabric. Tailored to an individual and manufactured.
'Concrete' / 'Wearable'?	No (a style has many items but a style as an abstract concept not wearable)	Closer but will argue 'no'. Slightly stretching the analogy will argue that 'flares' are a pattern for mens clothing for non-sporting activities	Yes

Architectural Styles

Architectural style: a specialization of elements and relation types, together with a set of constraints on how they can be used

[Clements and Kazman]

Architectural Styles

- Monolithic*
- Layered*
- Pipes and Filters*
- Database Centric*
- Service Oriented*
- Client-Server*
- REST*
- Message Driven
- Pub-Sub

** Again - many styles. Highlighted are ones I expect you to have an understanding of the architectural style*

Praising the Monolith

Monolithic is sometime used as a *derogatory* term. Monolith is a style with advantages and disadvantages. In many situations (particularly where complexity does not demand anything else) monolithic may be the best style.

Equally, engineers have microservices envy and apply in scenarios where it is not appropriate. Judgement!

* worshiped for its supposed magical powers or because it is considered to be inhabited by a spirit.

Which *style* is ‘best’?

Styles each have properties

- Advantages
- Disadvantages
- Trade-offs

Selection is based on *matching* desired quality attributes of the system we are building to the properties of styles

(we will cover this in architecture process)

Architectural Patterns

Architectural pattern: *“expresses a fundamental structural organization schema for software systems”*

[Buschmann et al 1996]

“It is above all a pattern, which in the context of architecture describes a particular recurring design problem that arises in specific design contexts, and presents a well proven generic scheme for its solution. The solution scheme is specified by describing its constituent components, their responsibilities and relations, and the ways in which they collaborate”

[Clemens]

Note: Style and Pattern are used ambiguously in industry. Layered is the style and 3-Tier is the pattern which is layered

Architectural *Patterns*

- Web App / 3-Tier*
- Microservices*

Reference Architecture

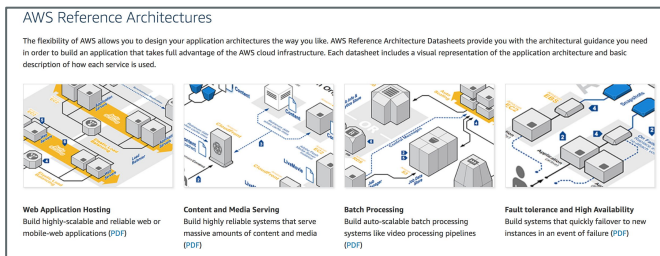
Reference Architecture: *a solution template for an architecture for a particular domain*

A reference architecture is more specific than a pattern as it can specify the *domain* and technology selection for elements

Reference Architectures

There are many *domain specific reference architectures*

- Data Science Platforms (emerging patterns & reference architectures)
- Data Pipelines (pipe and filter variant)
- [AWS Reference Architectures](#) (scroll down to reference architecture), [Horton](#), [MongoDB](#)



System : Styles/Patterns (1:1 or 1:m?)

- Is a system a single style or pattern?
- Or can a system include multiple styles or patterns?

More Specific (and specialized)

	Style	Pattern	Reference Architecture
Example	Layered	3-Tier/Web App	AWS/Beanstalk Web-App
Level of Abstraction	High	Moderate	Low
Specific Design Context	No (style applied to everything from internet to TCP to web apps)	Partial (web apps but nothing else)	Yes (web app on AWS or even more problem domain specific - financial trading system, media, telecoms)
Specifies Elements	Yes (though just 'layers')	Yes (layers are presentation, business, persist)	Yes (AWS EC2, ELB, Timeline, Order Mgmt, etc)
Specifies Relations	Yes	Yes (though may not specify exact technology)	Yes (AWS EC2, ELB, Timeline, Order Mgmt, etc)
Specifies Constraints	Yes ($L_N \rightarrow L_{>N}$)	Yes (presentation can only talk to middle tier etc. If pattern conforms to style then inherits style constraints ($L_N \rightarrow L_{>N}$))	Yes
Specifies Responsibilities of Elements	No (layers are layers)	Yes (presentation only has logic to present data handle interaction. Specific as to the responsibility of layer)	Yes
Specifies Technologies	No	No	Yes
Problem Domain Specific	No	No	Yes

Summary

- We do not need to create each architectures 'from scratch'
- Wealth of *styles*, *architectural patterns* and *reference architectures* to leverage
- Styles, patterns and R.A. are *different* in level of concreteness/abstraction
- Each style, pattern and R.A. has properties, pros and cons
- Understand problem domain and select appropriate style, pattern, R.A.

Pop Quiz

Question	Answer
The benefit of having established styles, patterns and reference architectures is ?	
The difference between an architectural style, pattern and reference architecture is {...}?	
The 'best' pattern to leverage for my needs is {...}? (carefull!)	
An example of the layered style is {.....}	

Reading

Reading	Optionality
Software Design Methodology : From Principles to Architectural Styles (chapter 6)	Required