



Product Family Management

Ken Jackson

www.qssinc.com

BUSINESS IS TOUGH.
TAKE CONTROL.





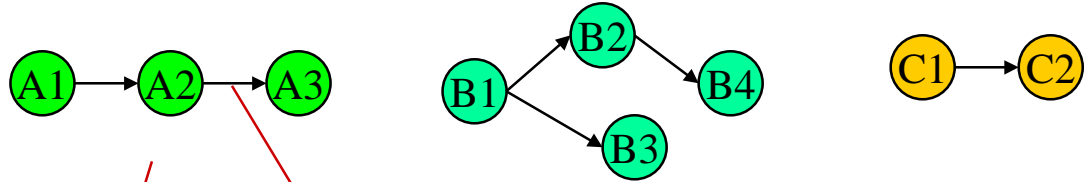
- Explain the problem
- Solution
- A simple example
- Summary of DOORS implementation
- Benefits summary



The problem....



Products

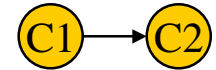
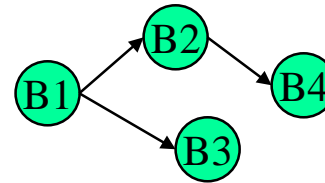


Versions and variants

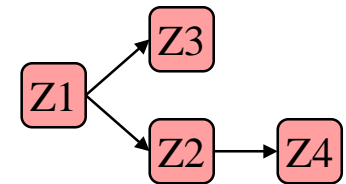
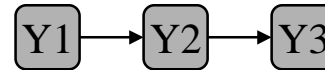
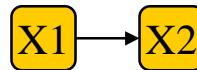
Need to manage relationships



Products



Sub-systems

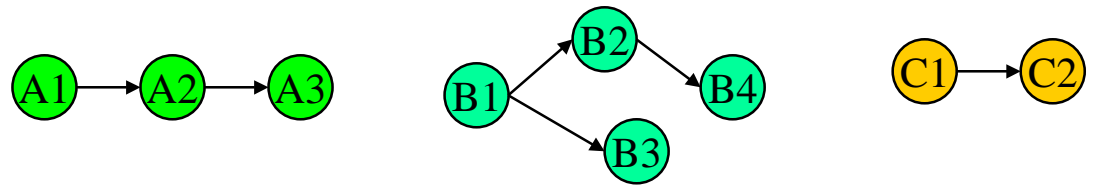


*Versions and
variants in
own life-cycle*

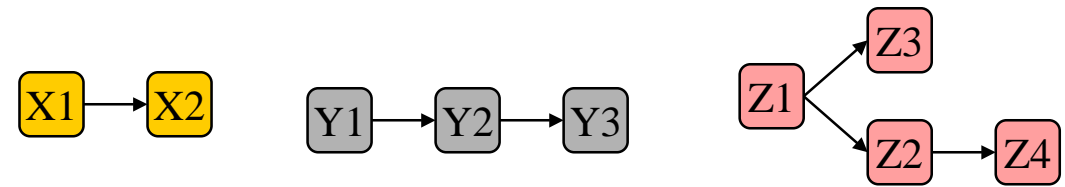
Almost anything can be an *item*



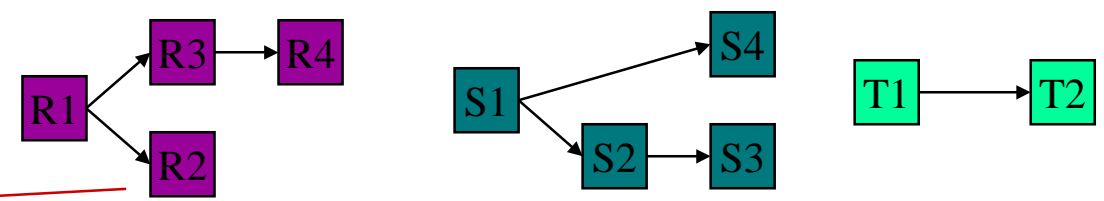
Products



Sub-systems



Components



Versions and variants in own life-cycle



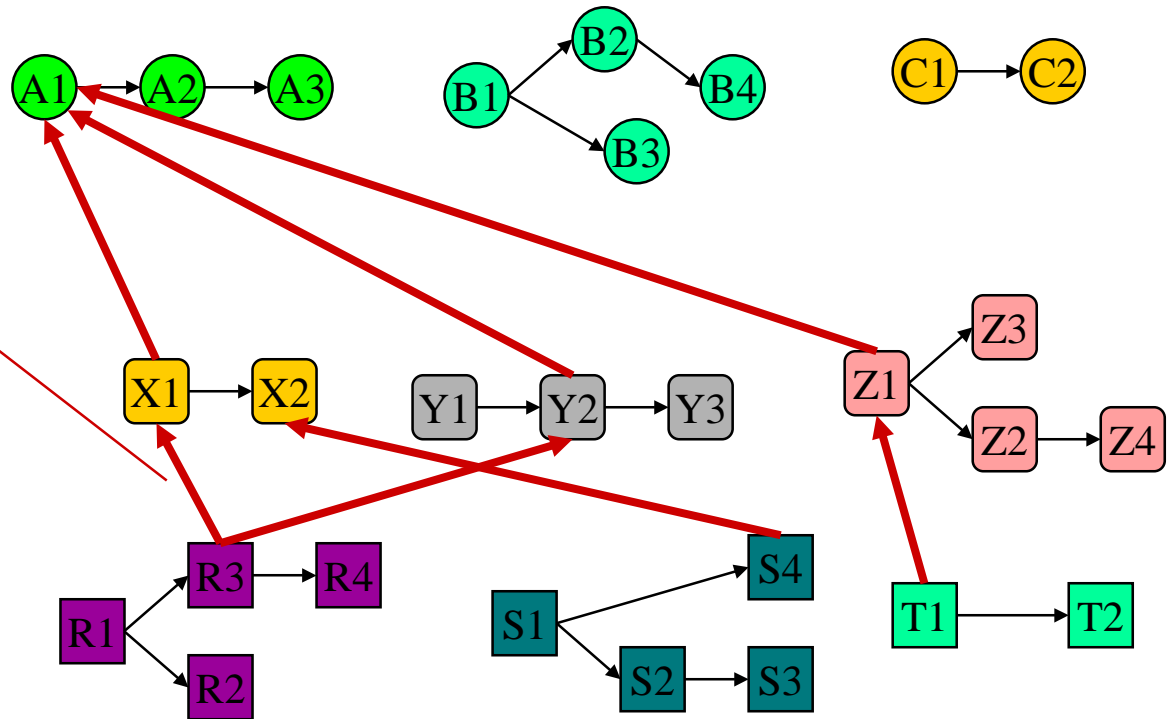
Product A1

Products

Sub

Components

Not strictly hierarchical



Different versions have different configurations



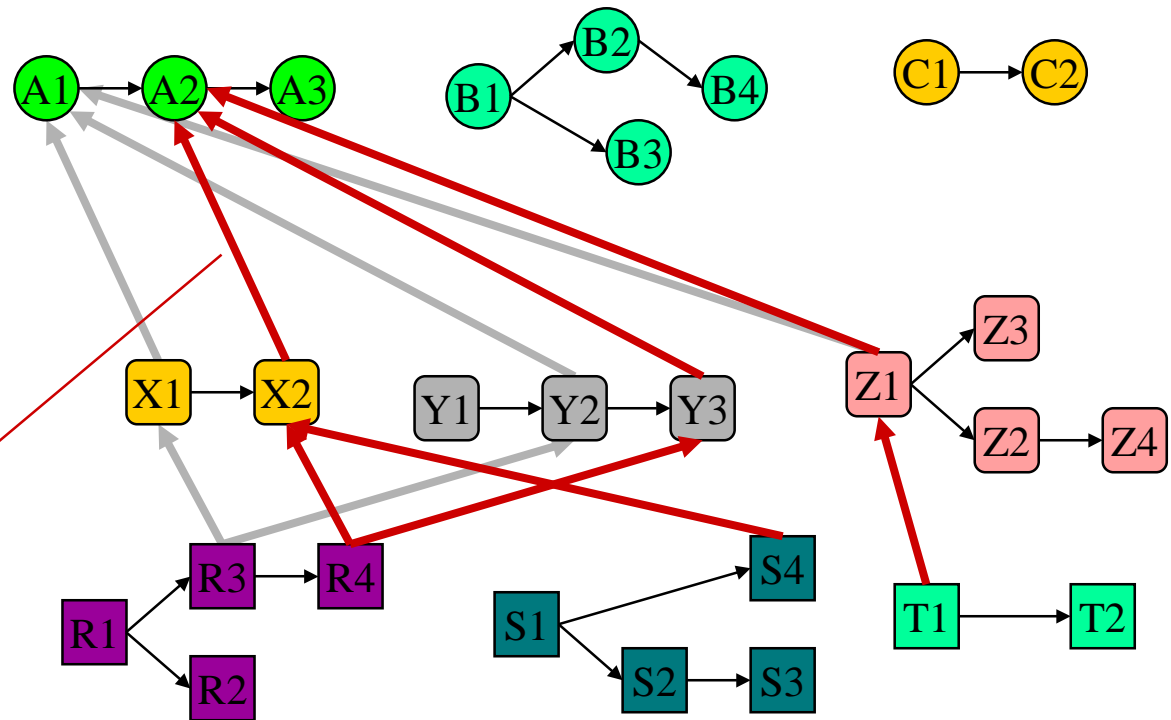
Product A2

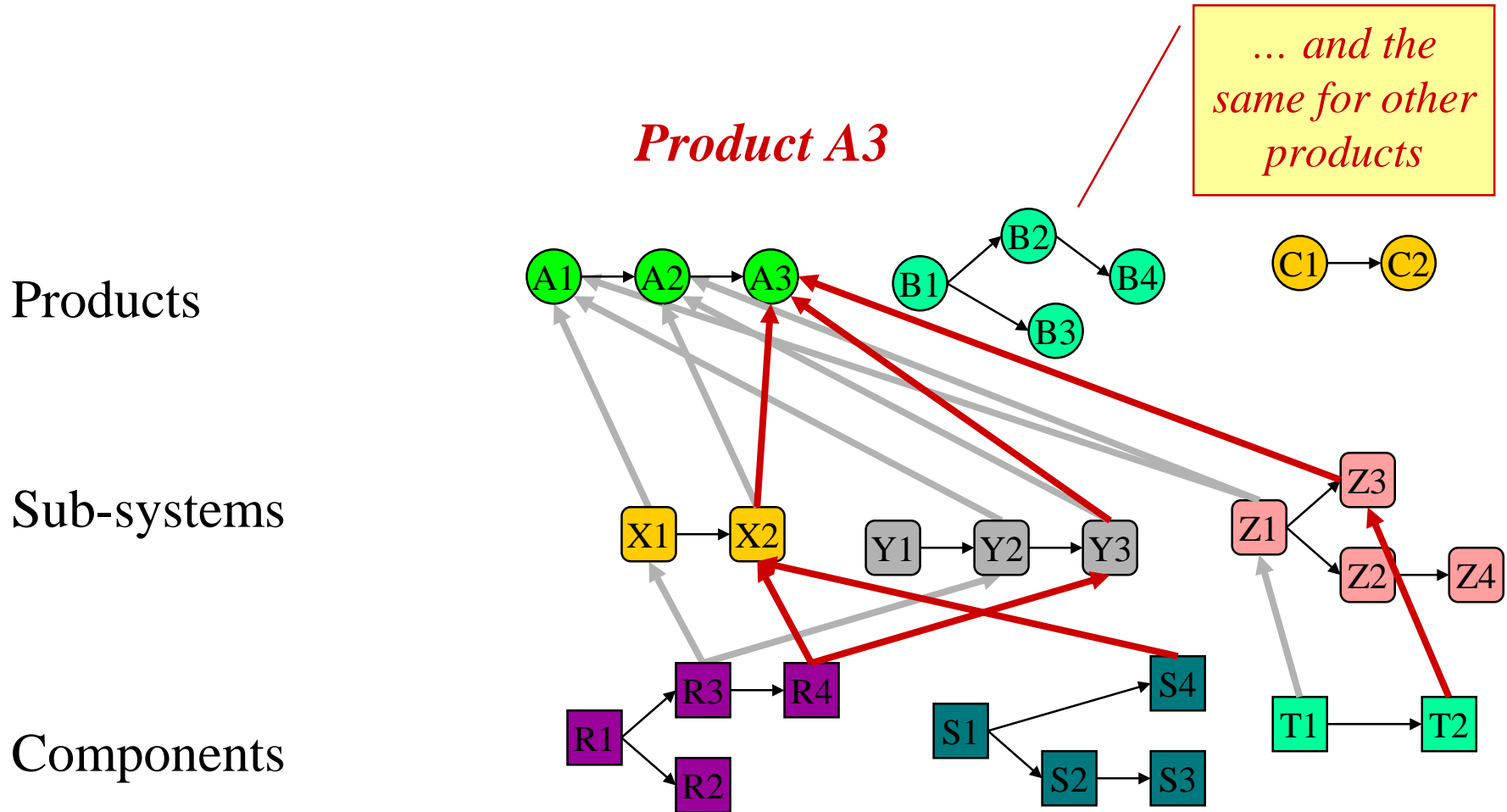
Products

Sub-systems

Co

*Need to
manage
evolution*





Common requirements across products



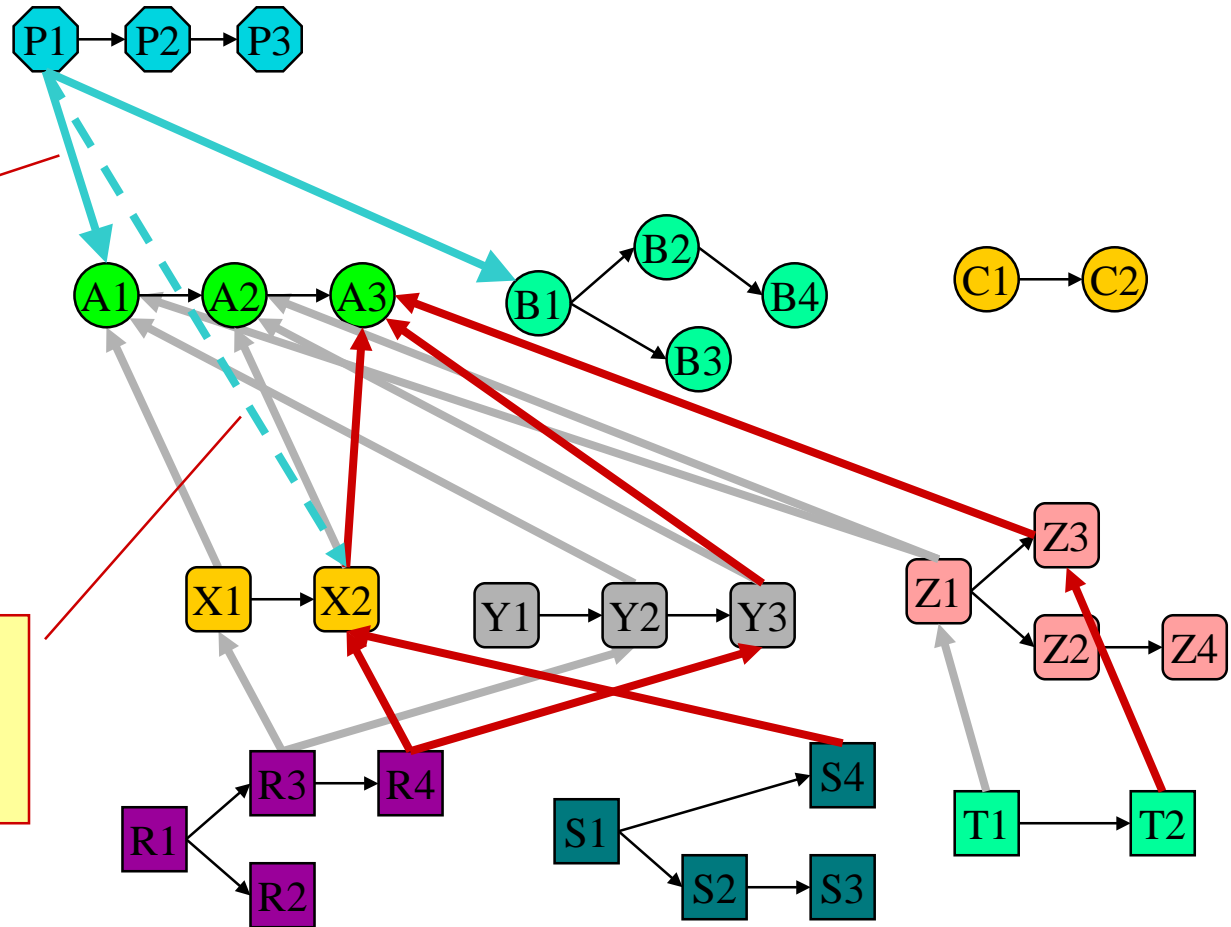
Common Requirements

Common requirements "inherited"

Sub-systems

Should not be necessary

Components



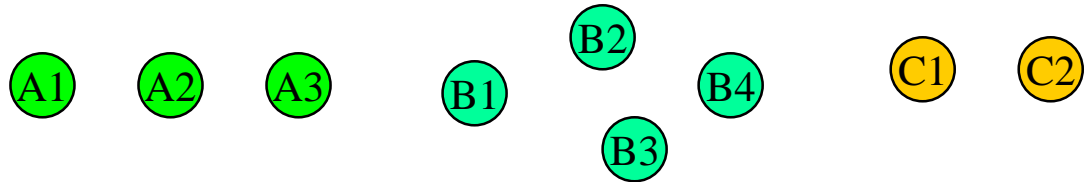


- Versions and variants of:
 - **single items**

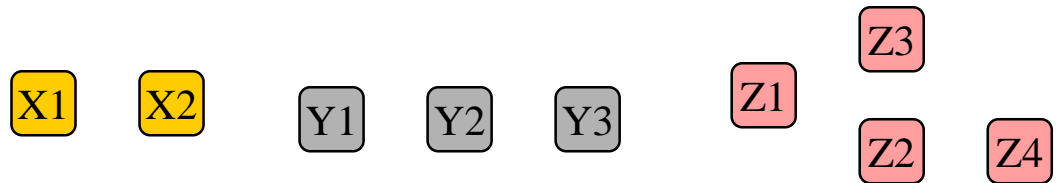


1. Use a DOORS Module for each version/variant of an item

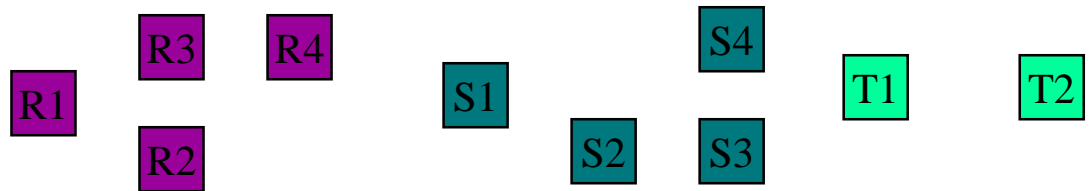
Products



Sub-systems

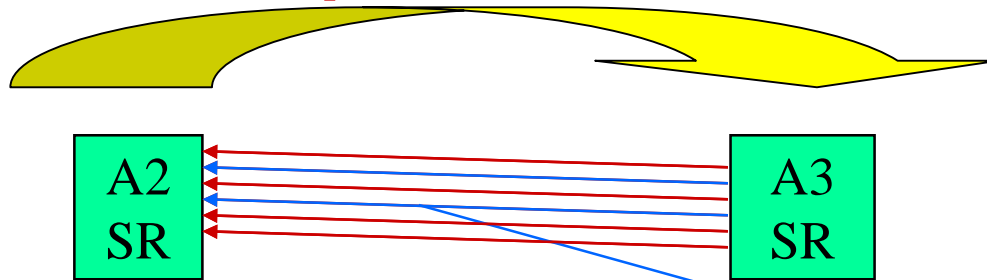


Components





1. Duplicate the module



2. Link every requirement to the original.

3. Mark every requirement as inherited.

4. Specialise requirements for the new version



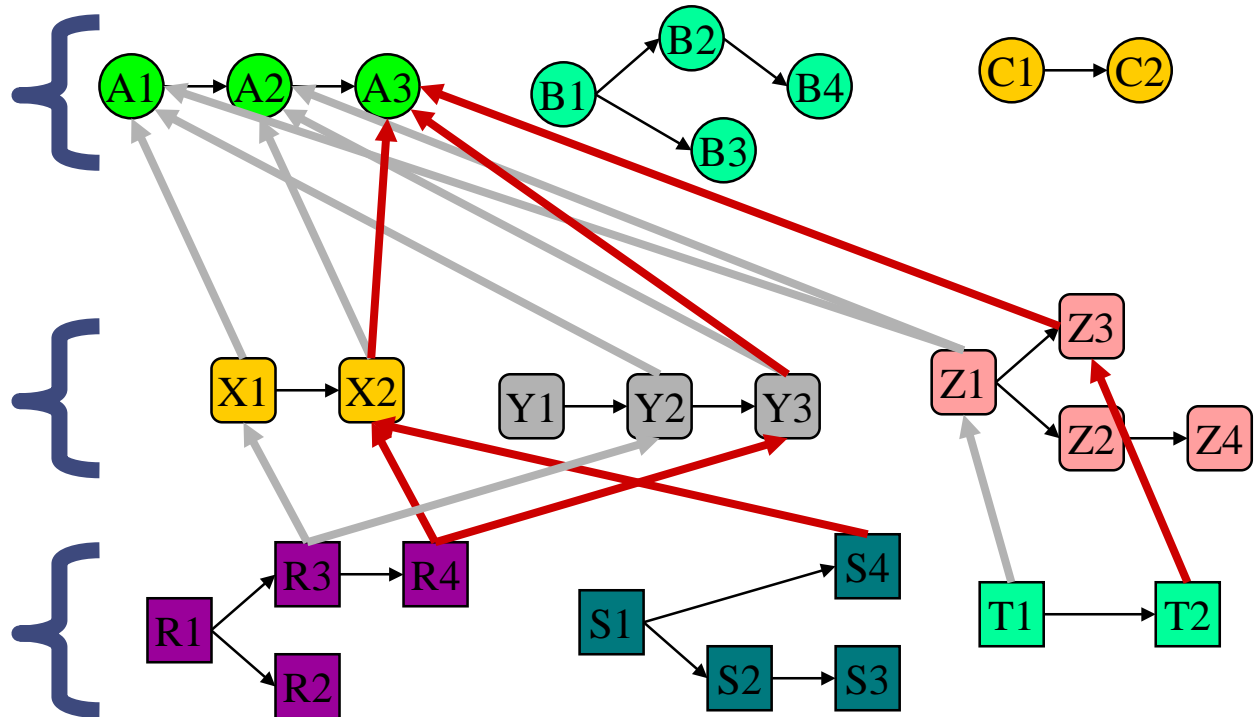
- Versions and variants of:
 - **composites**
- Main concept:
 - **layer index**



Layer 1 = Products

Layer 2 = Sub-systems

Layer 3 = Components



*Items from
one layer
map to
lower layer(s)*

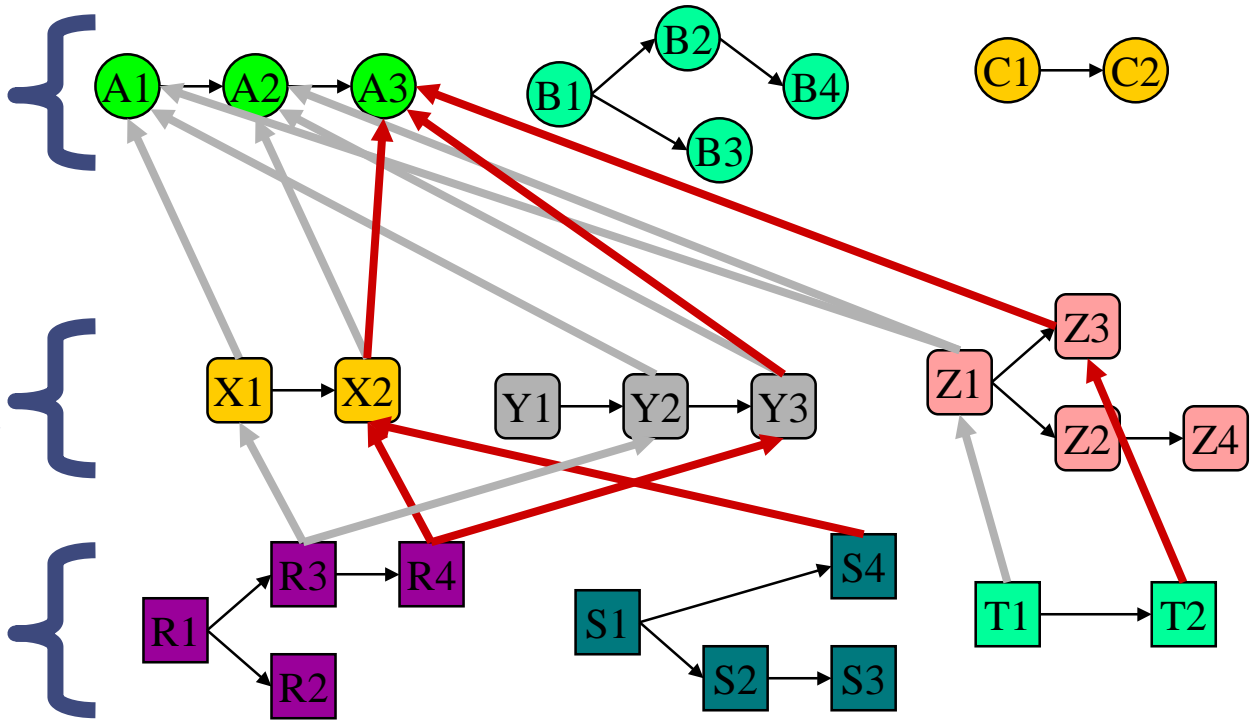
Configuration index(es)



Layer 1 = Products

Layer 2 = Sub-systems

Layer 3 = Components



Sub-systems

	X	Y	Z	p	q	r
X1	X1	Y1	Z1			
X2	X2	Y2	Z2			
X3	X3	Y3	Z3			
		Y2		Y2		
				Y2		
				Y2		
				Y2		
X3						

Products

Components

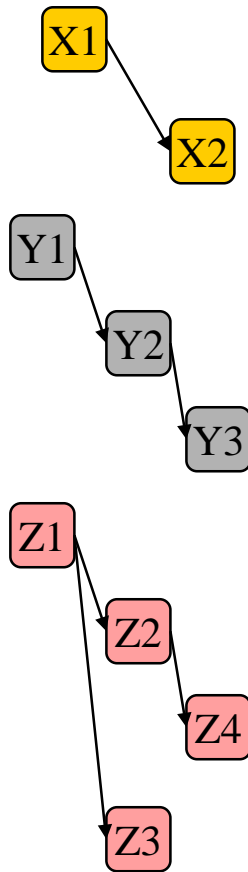
	R	S	T	u	v	w
X1	R1					
X2	R2					
X3	R3					
		S2				
		S3				
			T1			
X3						

Sub-systems



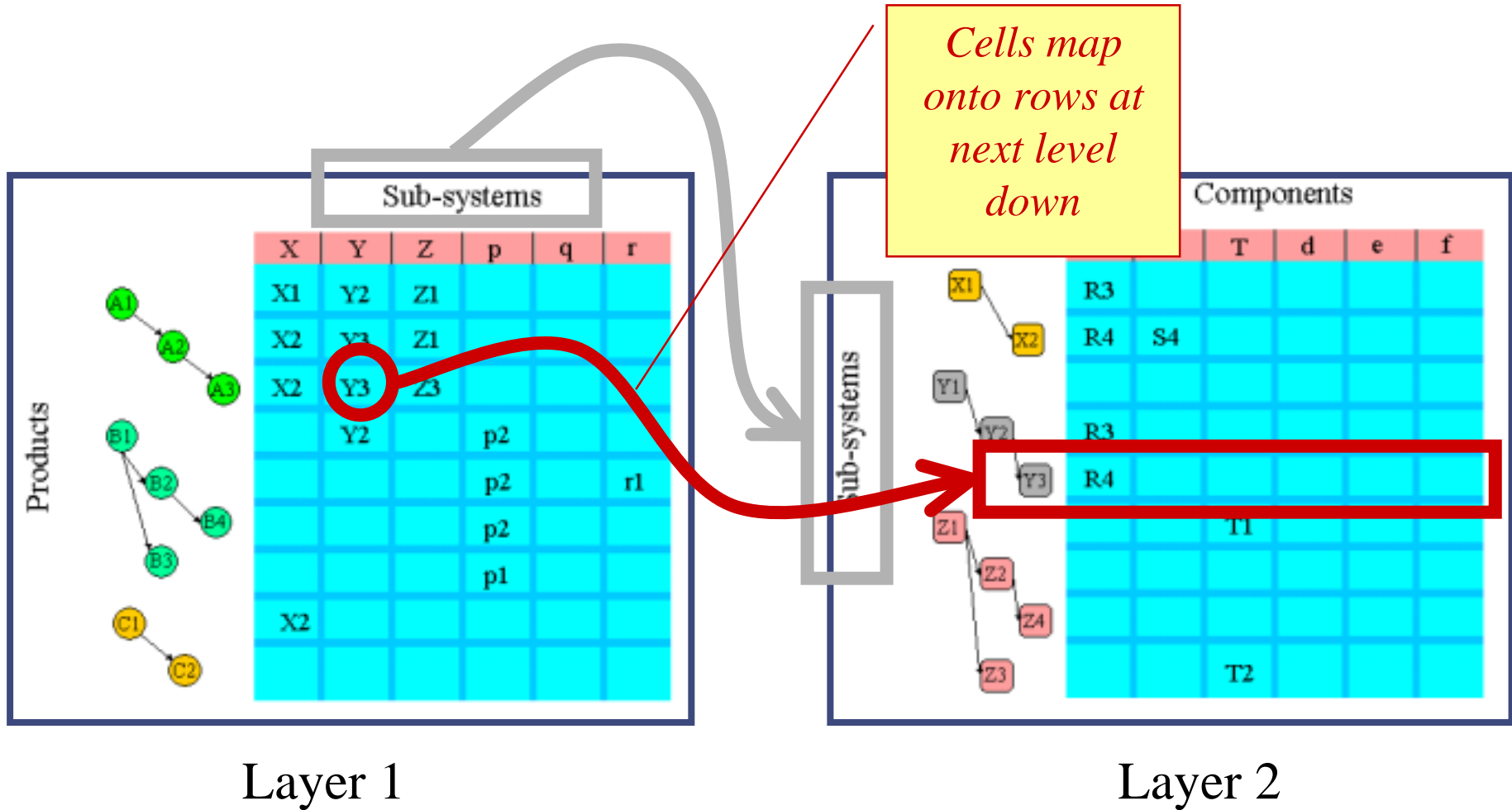
Components

Sub-systems

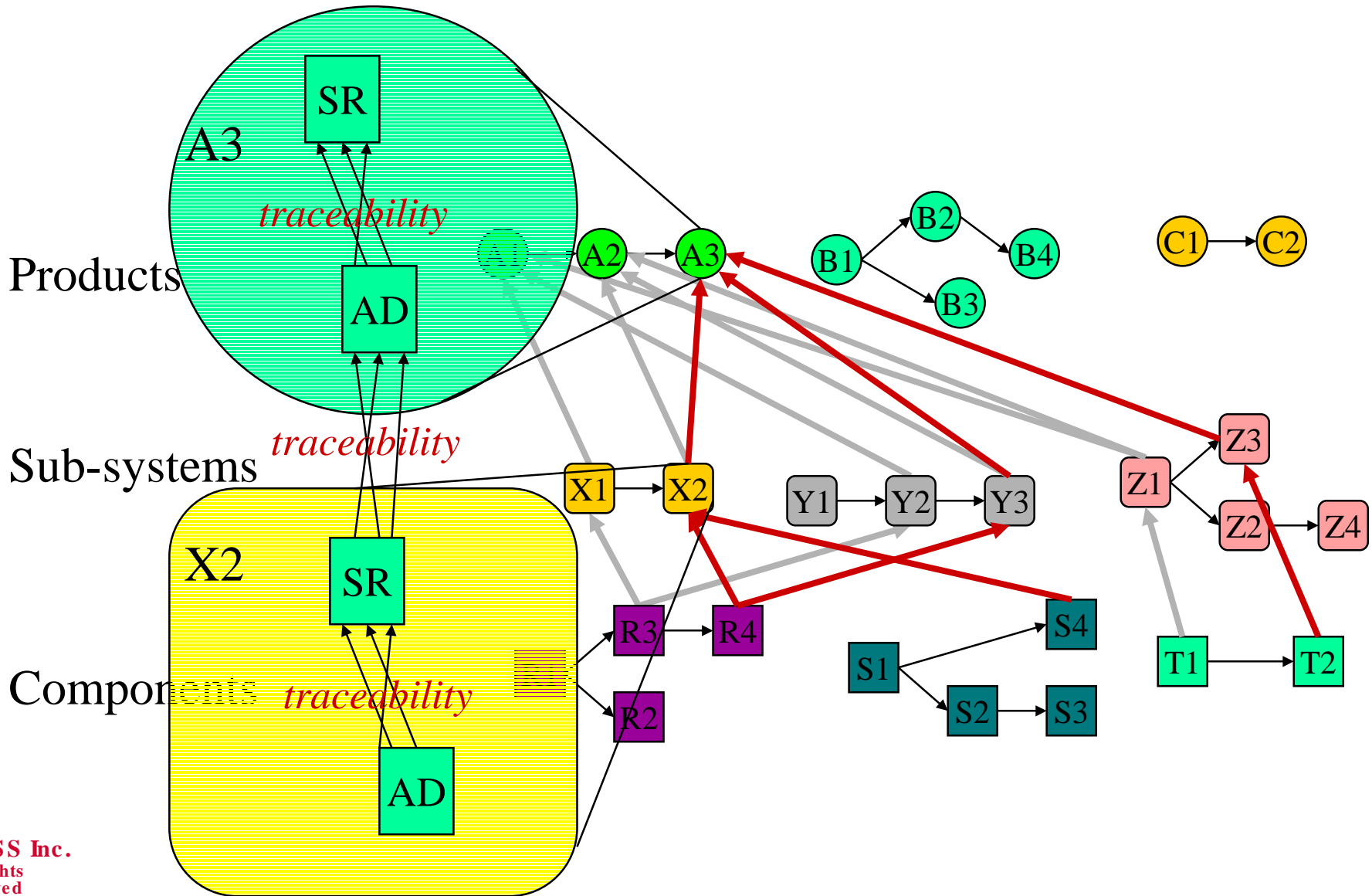


R	S	T	d	e	f
R3					
R4	S4				
R3					
R4					
		T1			
		T2			

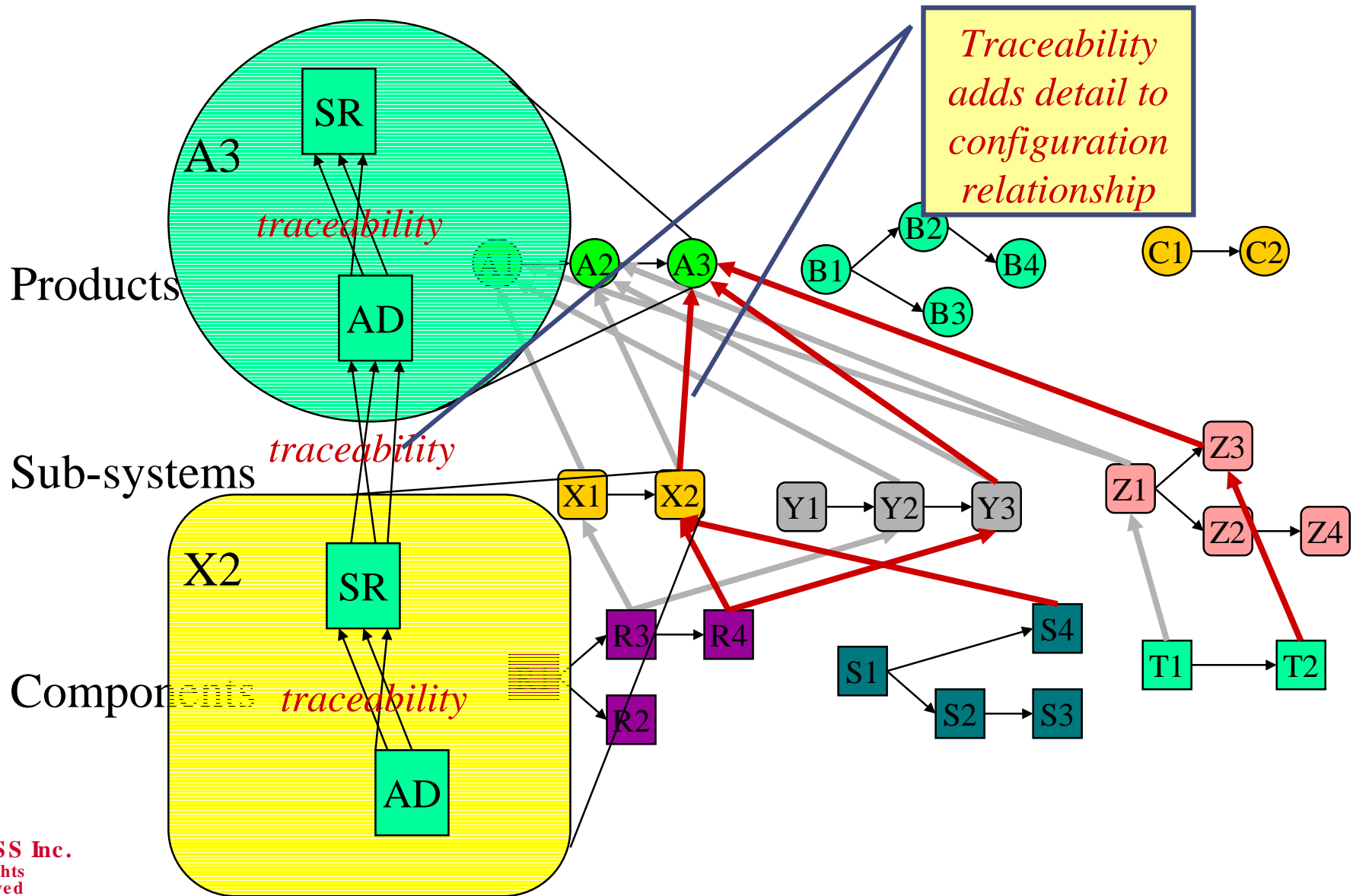
Relationship between layers

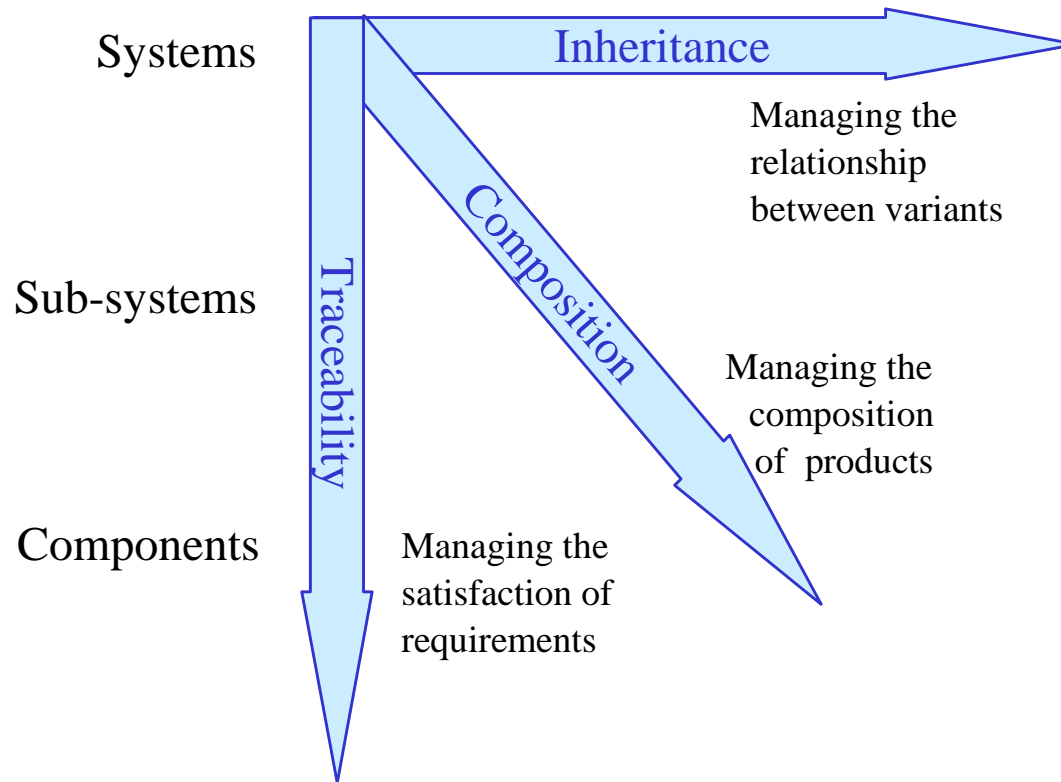


Traceability



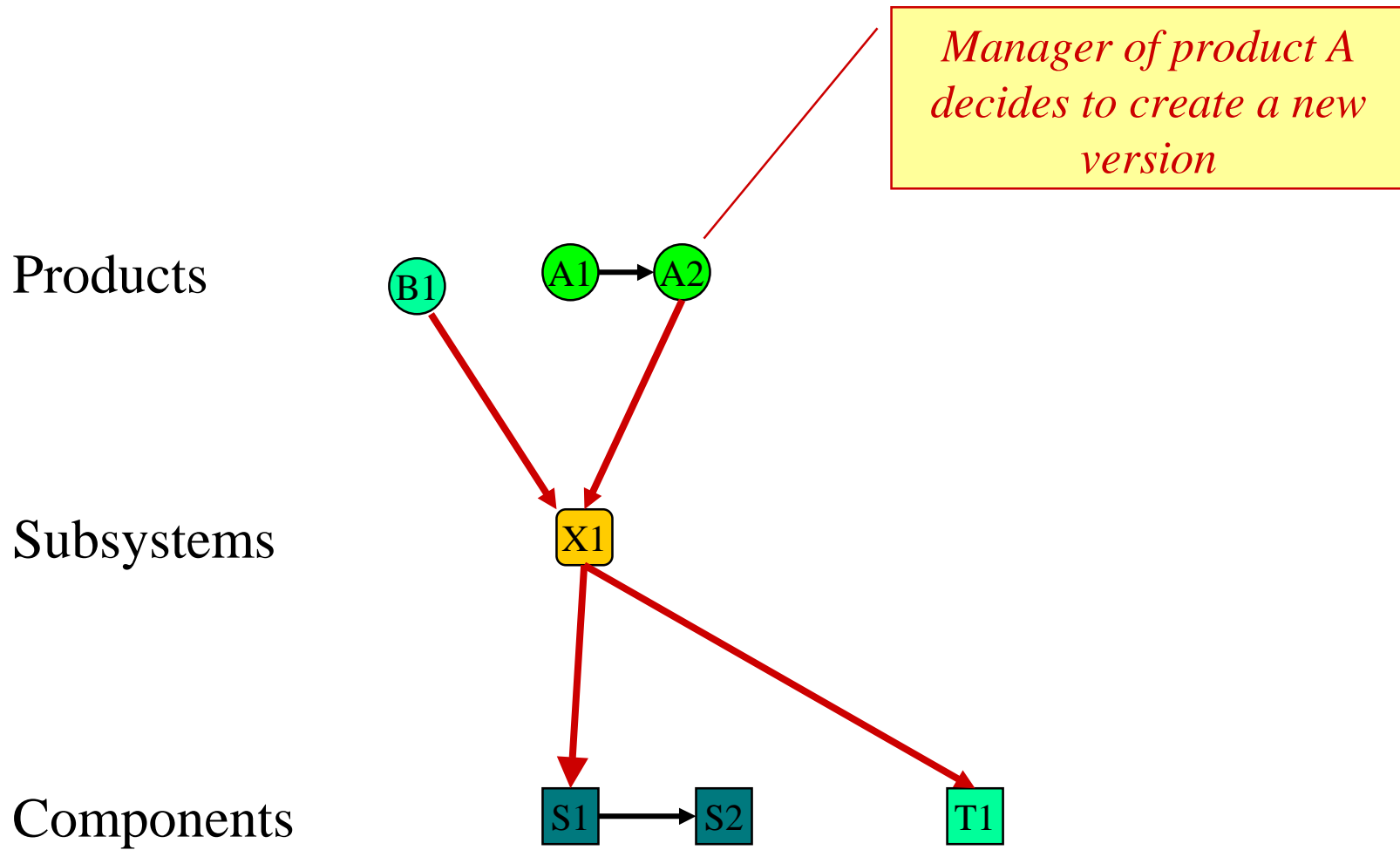
Traceability v. configuration

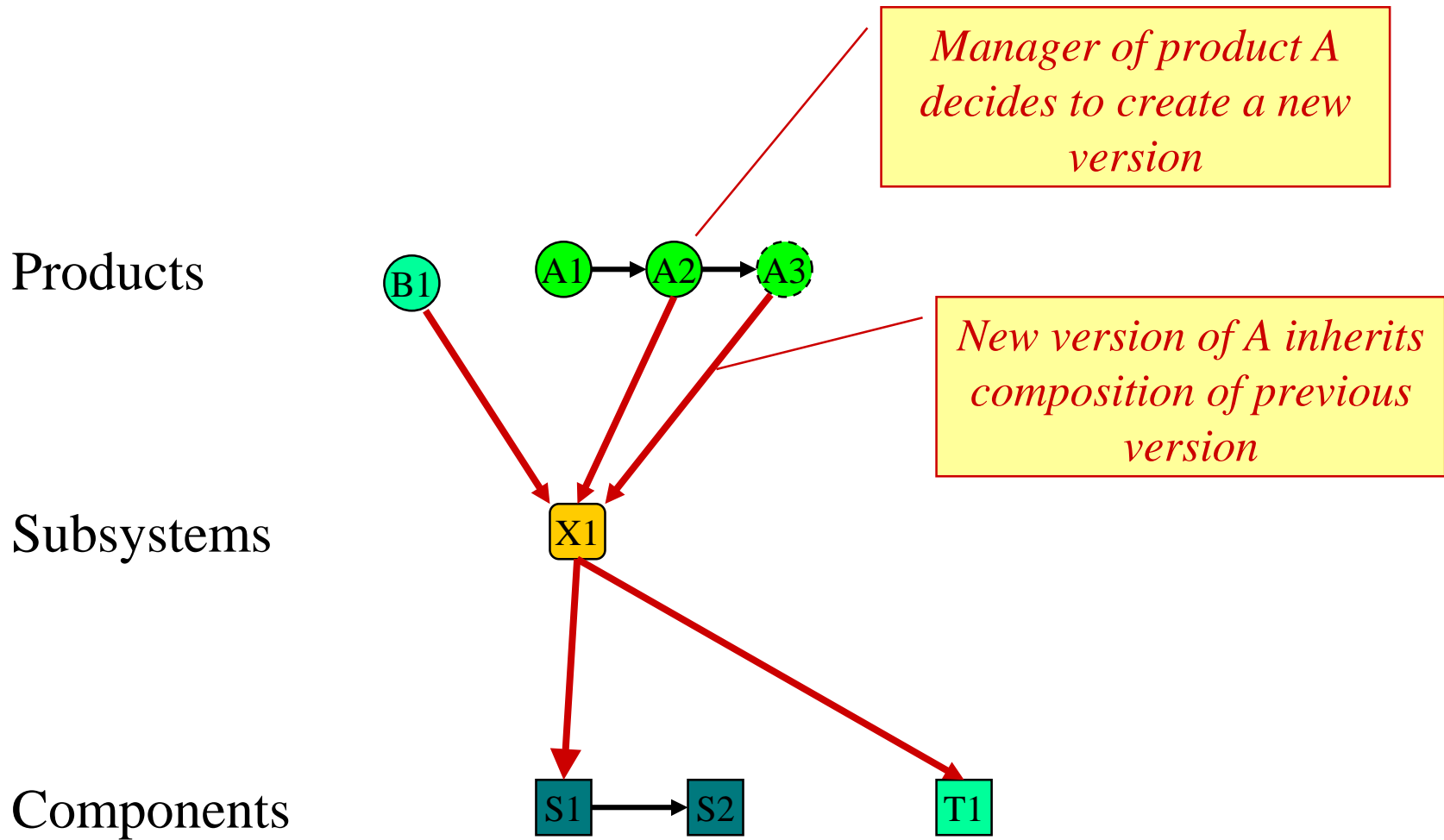






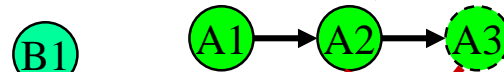
A Simple Example





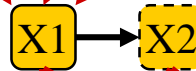


Products



*Manager of product A
decides to create a new
version*

Subsystems



*New version of A inherits
composition of previous
version*

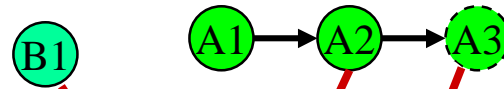
Components



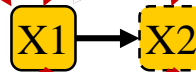
*Subsystem supplier
decides that a new
version of X will be
necessary*



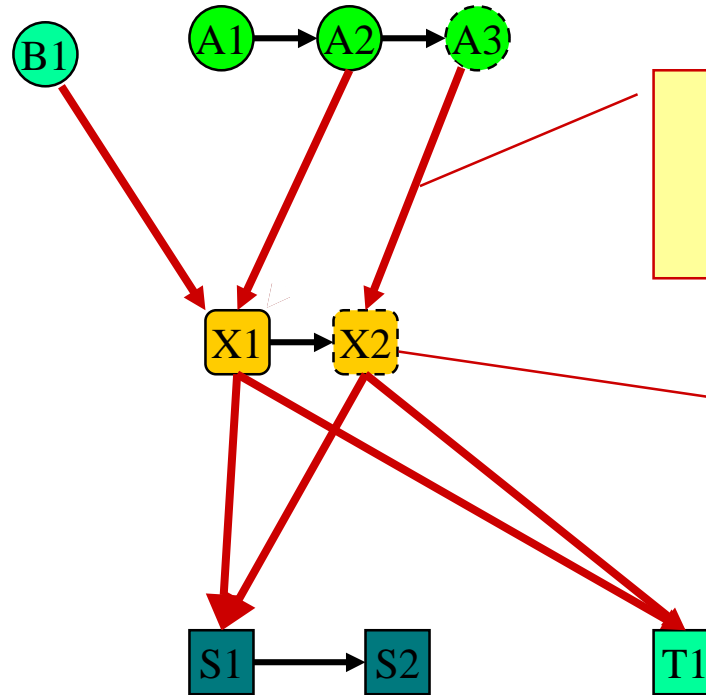
Products



Subsystems



Components



*Subsystem supplier
requests new
composition*

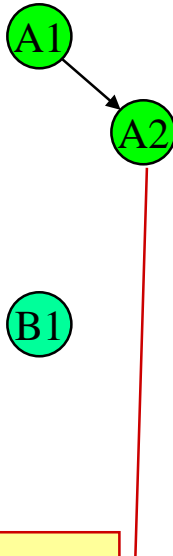
*Subsystem supplier
decides that a new
version of X will be
necessary*



Sub-systems

X	Y	Z	p	q	r
X1					
X1					
X1					

Products



*New version
(A3) of product
A starts from A2
base*



Implementation in DOORS

Solution 1: In DOORS

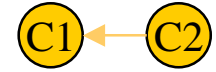
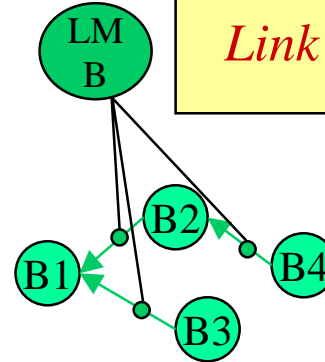


Each version/variant is a DOORS Formal Module

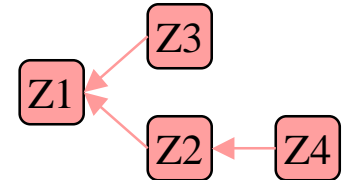
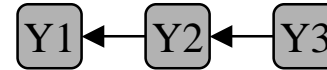
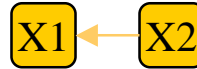
Each version/variant has link set to predecessor Formal Module

Each item has its own Link Module

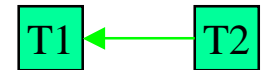
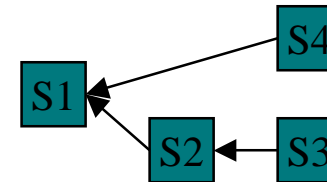
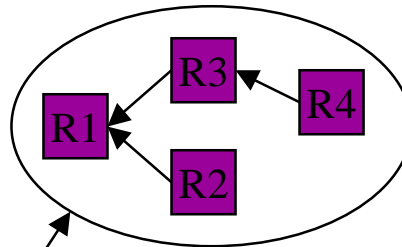
Products



Sub-systems



Components



Versions and variants in own life-cycle per item

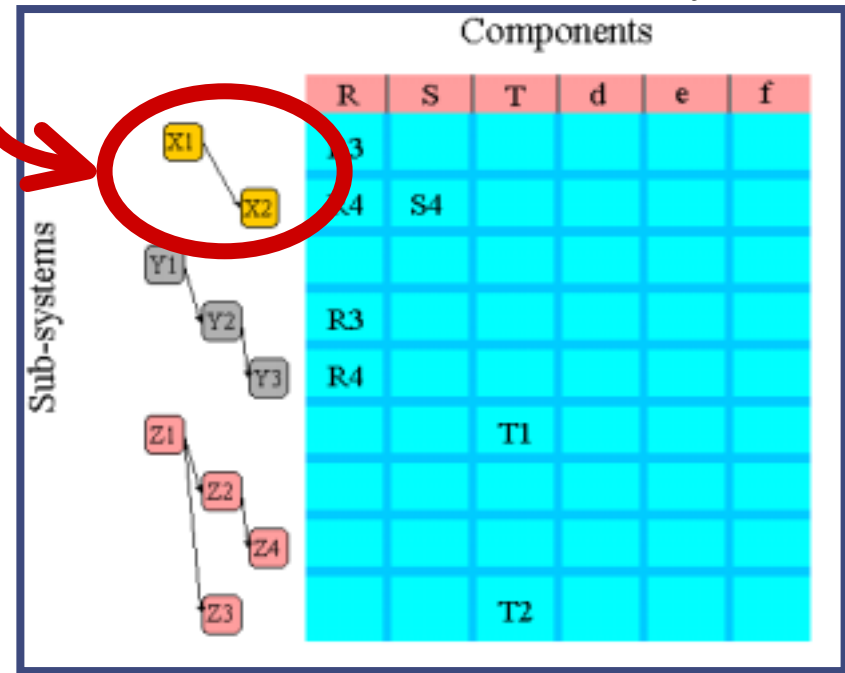
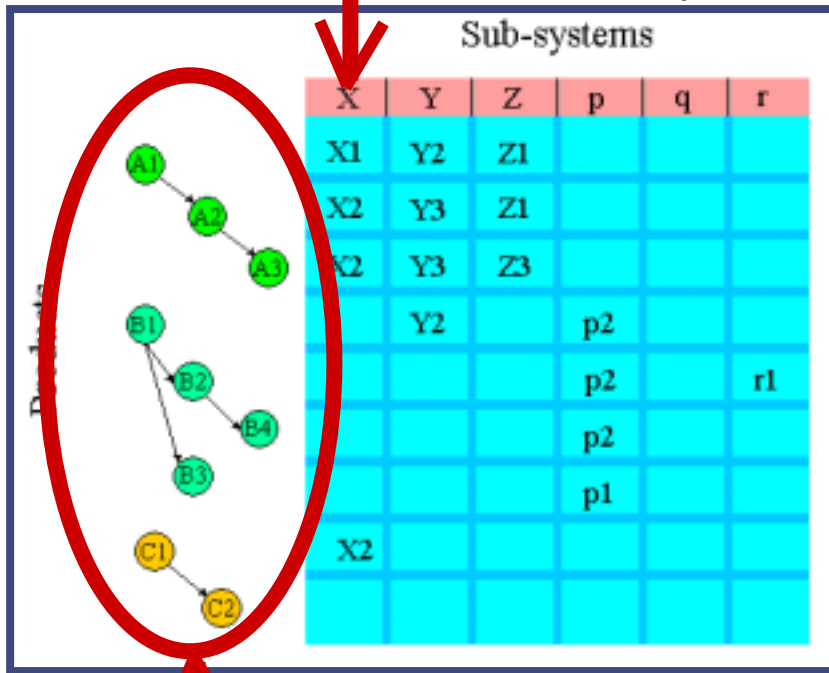
Solution 2: In DOORS



Enumeration Attribute derived from

Layer 1

Layer 2



Objects in hierarchy = precedence

Layer Index view in DOORS Formal Modules



- Version/Variant represented by (set of) DOORS modules
- Inheritance by copy and link
- Specialisation by modifying link type
- Composition organised via special module at each layer
- Solution is feasible
- Solution can be enhanced and policed by special purpose DXL
- Visualisation of configuration links in addition to configuration index



- Combines configuration management with requirements management
- Understands relationship between traceability and configuration management
- Built-in reuse through requirements inheritance
- DOORS 5 provides convenient organisation
- Better with some DXL to police naming conventions
- Paper presented at Technical Management conference in Dallas, Texas September 2000