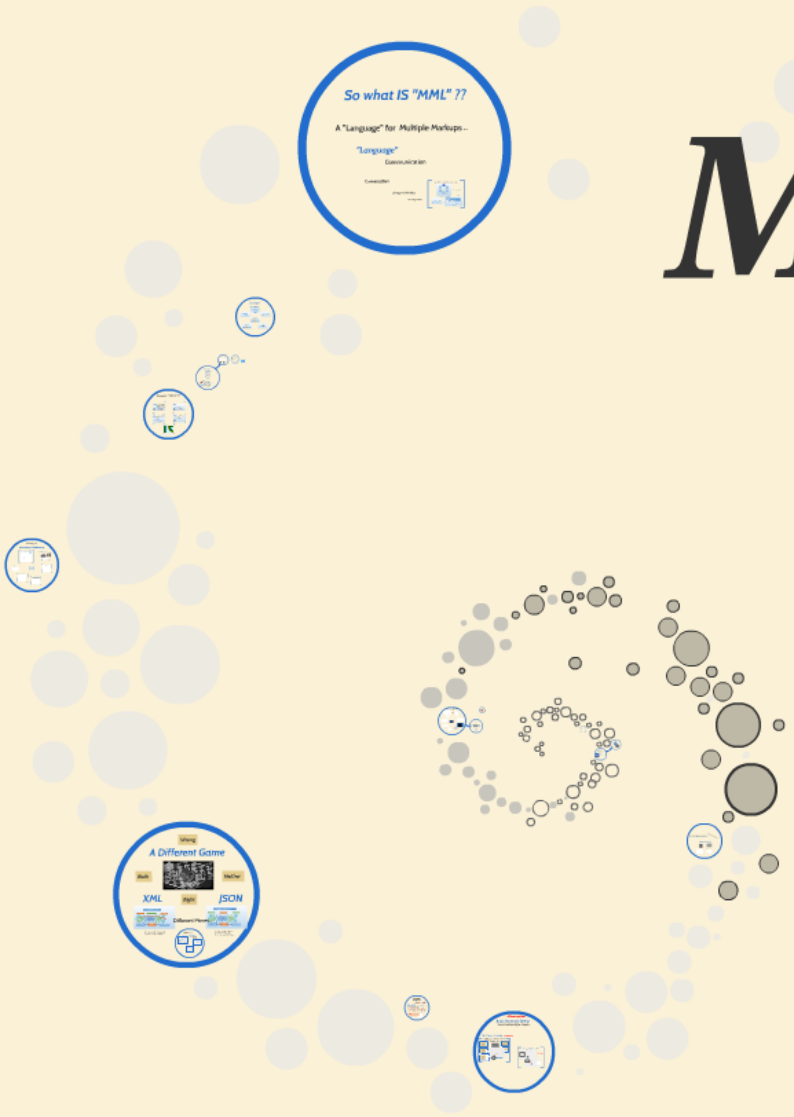
MML



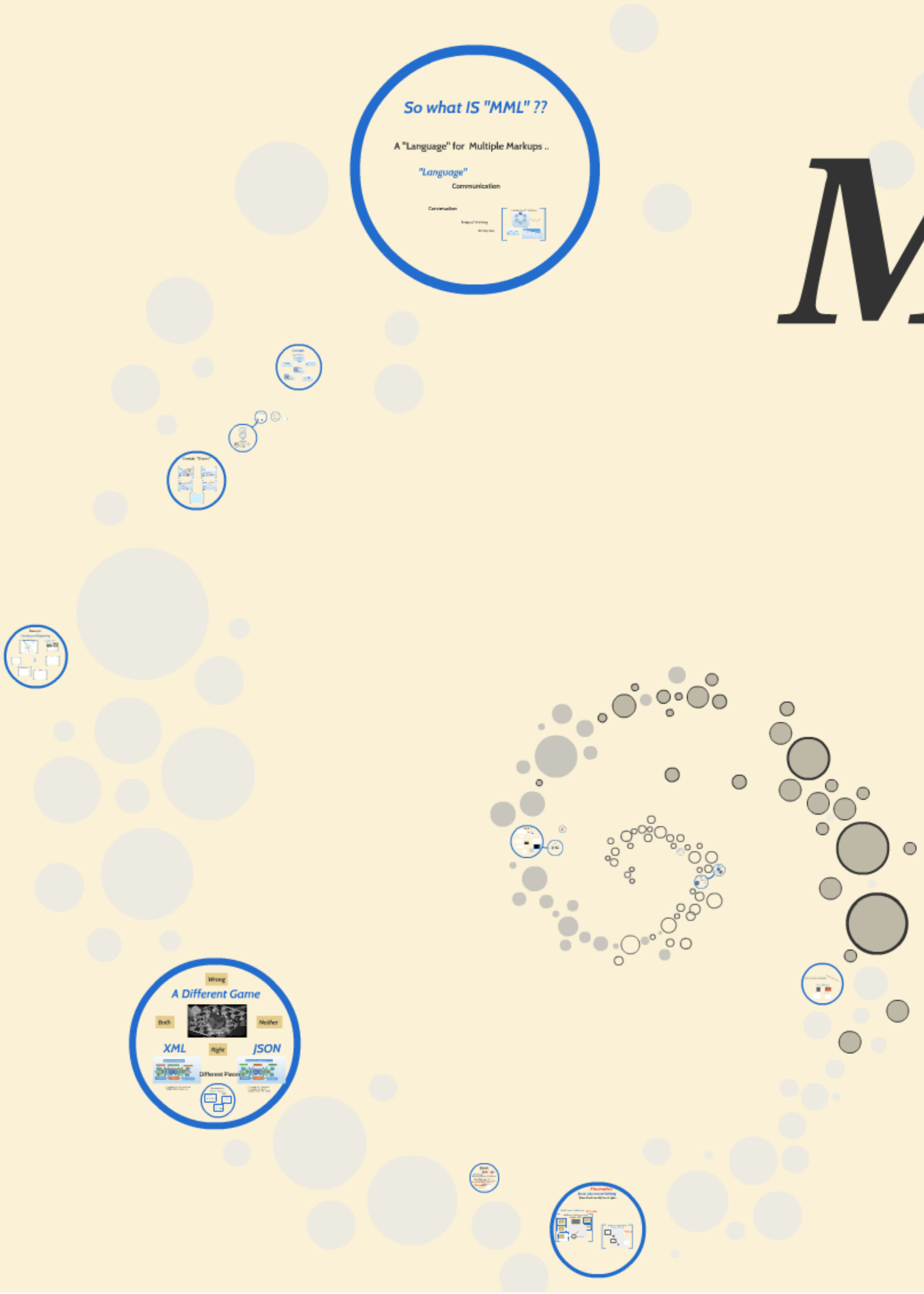
MultiMarkup Language

When *one "markup"* just isn't enough...

MML

MultiMarkup Language

When *one "markup"* just isn't enough...



So what IS "MML" ??

A "Language" for Multiple Markups ..

"Language"

Communication

A Different Game

MML

MultiMarkup Language

When *one "markup"* just isn't enough...

Dead Markup

I've seen the dead many times

Dead Markup

I've seen the dead many times

"Simplicity"

"Language"

"Assumptions"

-- James Clark -- on RNG

> "
> *Simplicity of specification often goes hand in hand with simplicity of use. But I find that these are often in conflict with simplicity of implementation.*
> "

Balisage 2015
The Markup Conference

There is nothing so practical as

Something that WORKS!!!

How to Milk a Cow?

I have a Good Theory !!!

I Want MILK !!!!

Balisage 2015
The Markup Conference

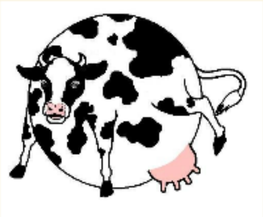
There is nothing so practical as

Something that WORKS!!!

How to Milk a Cow?

I have a Good Theory !!!

I Want MILK !!!!

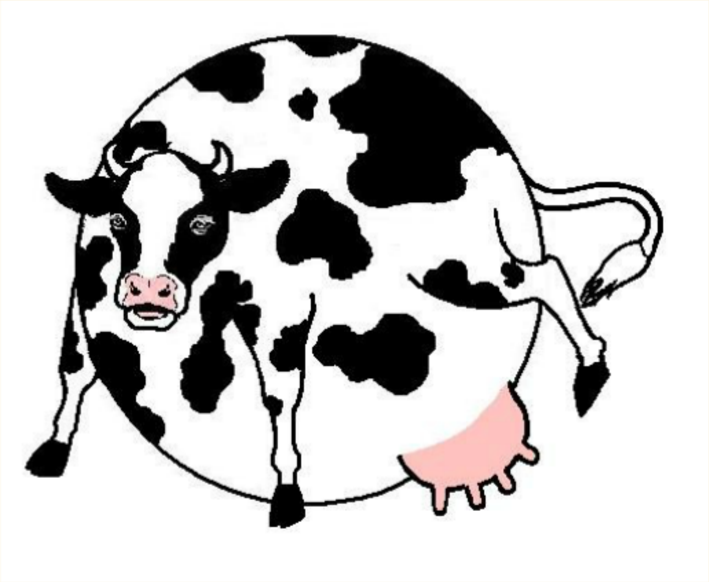


While you refine your theory ...

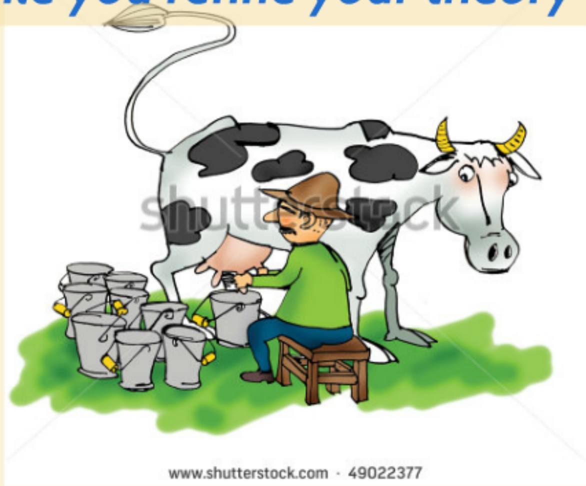
How to Milk a Cow?

I have a Good Theory !!!

I Want MILK !!!!



While you refine your theory ...



What Is ...

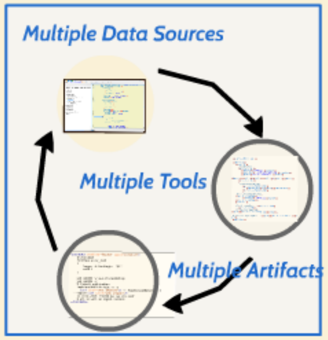
"MML"

Multiple Markup *Language*

A *Language* for managing Multiple Markup Languages

Hierarchical and Multi-Normative

Why would anyone want to do that ?



Because Sometimes ...

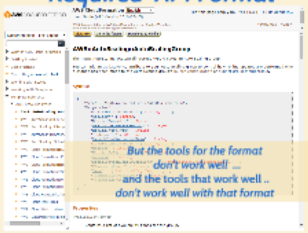
Your "Document" contains "Data"

The "Data" contains "Documents"

Which contain "Data"
- *and "Text"*
- *and "Documentation"*
- *and "Software"*
- *and "Scripts"*
- *and "Provisioning"*
- *and "Logic"*

but no "Markup"

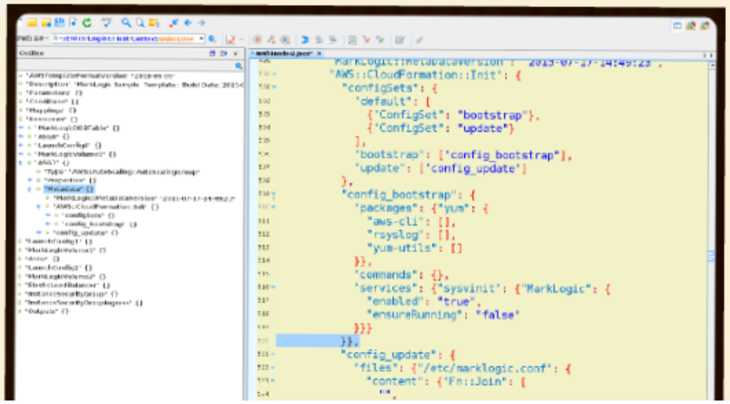
Required API Format



But the tools for the format don't work well ...
and the tools that work well ..
don't work well with that format

Its more fun to make tools then use them

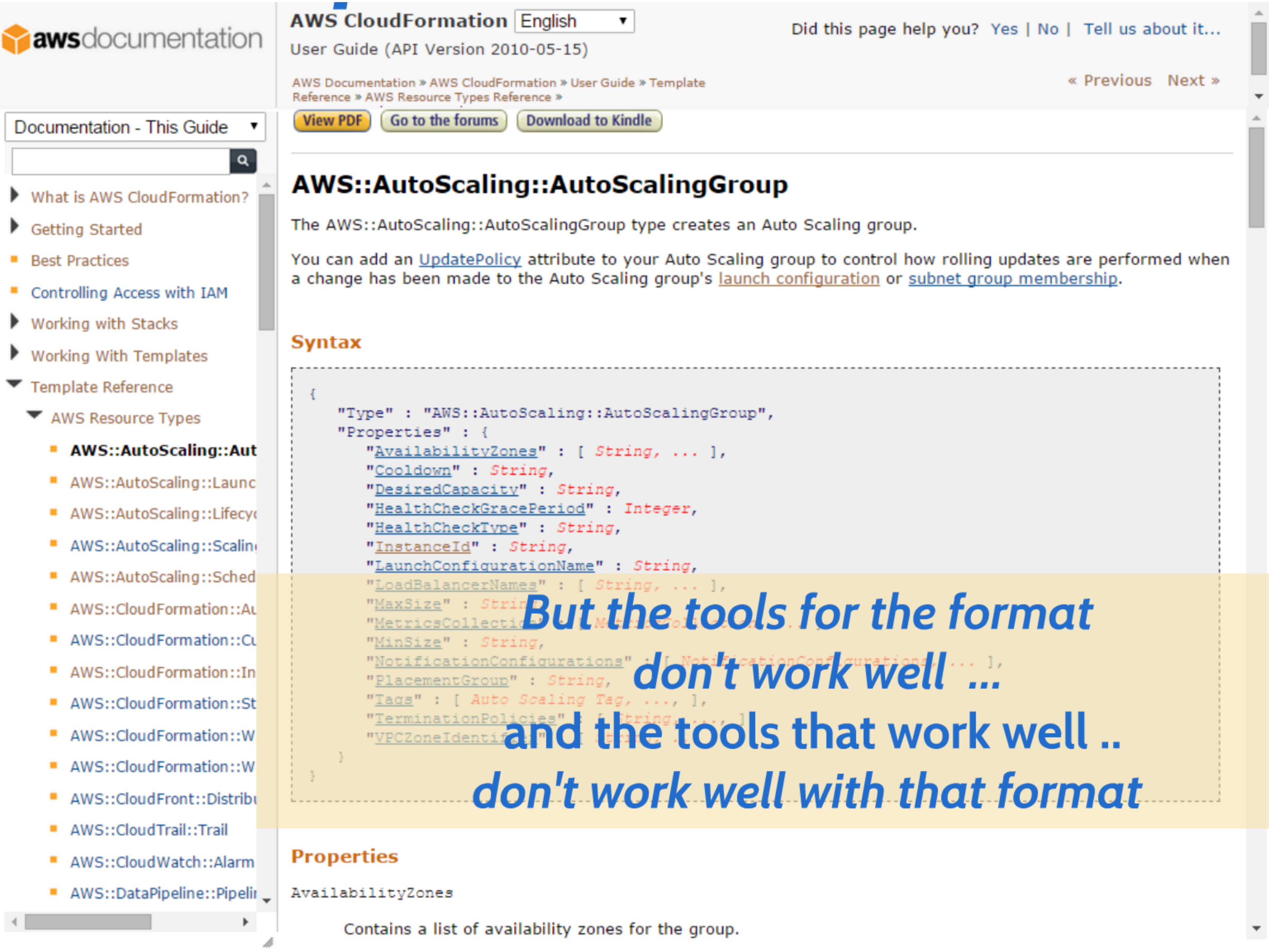
Multiple Data Sources



Outline

```
'AWSTemplateFormatVersion' "2010-09-09"
'Description' 'MarkLogic Sample Template: Build Date: 2015-0
'Parameters' {}
'Conditions' {}
'Mappings' {}
'Resources' {}
  'MarkLogicDDBTable' {}
  [illegible] {}
  'LaunchConfig1' {}
  'MarkLogicVolume1' {}
  'ASG1' {}
    "Type" "AWS::AutoScaling::AutoScalingGroup"
    "Properties" {}
    "Metadata" {}
      "MarkLogic::MetadataVersion" "2015-07-17-14:49:23"
      'AWS::CloudFormation::Init' {}
        'configSets' {}
        'config_bootstrap' {}
      'config_update' {}
'LaunchConfig1' {}
'MarkLogicVolume1' {}
'ASG2' {}
'LaunchConfig2' {}
'MarkLogicVolume2' {}
'ElasticLoadBalancer' {}
'InstanceSecurityGroup' {}
'InstanceSecurityGroupIngress' {}
'Outputs' {}
```

```json
      "MarkLogic::MetadataVersion": "2015-07-17-14:49:23",
      "AWS::CloudFormation::Init": {
        "configSets": {
          "default": [
            {"ConfigSet": "bootstrap"},
            {"ConfigSet": "update"}
          ],
          "bootstrap": ["config_bootstrap"],
          "update": ["config_update"]
        },
        "config_bootstrap": {
          "packages": {"yum": {
            "aws-cli": [],
            "rsyslog": [],
            "yum-utils": []
          }},
          "commands": {},
          "services": {"sysvinit": {"MarkLogic": {
            "enabled": "true",
            "ensureRunning": "false"
          }}}
        }},
        "config_update": {
          "files": {"/etc/marklogic.conf": {
            "content": {"Fn::Join": [
              "",
```



AWS::AutoScaling::AutoScalingGroup

The AWS::AutoScaling::AutoScalingGroup type creates an Auto Scaling group.

You can add an UpdatePolicy attribute to your Auto Scaling group to control how rolling updates are performed when a change has been made to the Auto Scaling group's launch configuration or subnet group membership.

Syntax

```
{
   "Type" : "AWS::AutoScaling::AutoScalingGroup",
   "Properties" : {
      "AvailabilityZones" : [ String, ... ],
      "Cooldown" : String,
      "DesiredCapacity" : String,
      "HealthCheckGracePeriod" : Integer,
      "HealthCheckType" : String,
      "InstanceId" : String,
      "LaunchConfigurationName" : String,
      "LoadBalancerNames" : [ String, ... ],
      "MaxSize" : String,
      "MetricsCollection" : [ MetricsCollection, ... ],
      "MinSize" : String,
      "NotificationConfigurations" : [ NotificationConfigurations, ... ],
      "PlacementGroup" : String,
      "Tags" : [ Auto Scaling Tag, ..., ],
      "TerminationPolicies" : [ String, ..., ],
      "VPCZoneIdentifier" : [ String, ... ]
   }
}
```

Properties

AvailabilityZones

Contains a list of availability zones for the group.

But the tools for the format
don't work well ...
and the tools that work well ..
don't work well with that format

Your "Document" contains "Data"

The "Data" contains "Documents"

Which contain "Data"

- *and "Text"*
- *and "Documentation"*

Stick to One Markup

"Easy" !!! ...

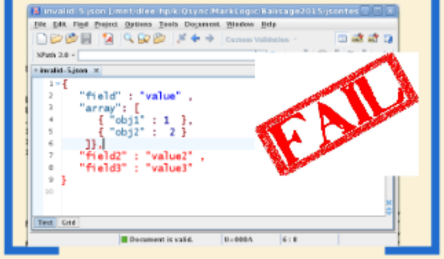
Just convert to XML

If only there were ...

A supported implementation ...
A small fixed set of elements and types ..
A Schema -- of any sort
An API that doesn't change
Formal Semantics and validation
Your software is perfect ever time
Tools and People to help ...

Just Use JSON ..

The "Simple" things are the hardest



```
{
   "field" : "value" ,
   "array": [
      { "obj1" : 1  },
      { "obj2" :  2 }
   ]},
   "field2" : "value2" ,
   "field3" : "value3"
}
```

invalid-5.json [/mnt/diee-hp/k/Qsync/MarkLogic/Balisage2015/jsontes
File Edit Find Project Options Tools Document Window Help
Custom Validation
XPath 2.0
invalid-5.json
Text Grid
Document is valid.
U+000A
6 : 8
FAIL

... If the real cow was spherical

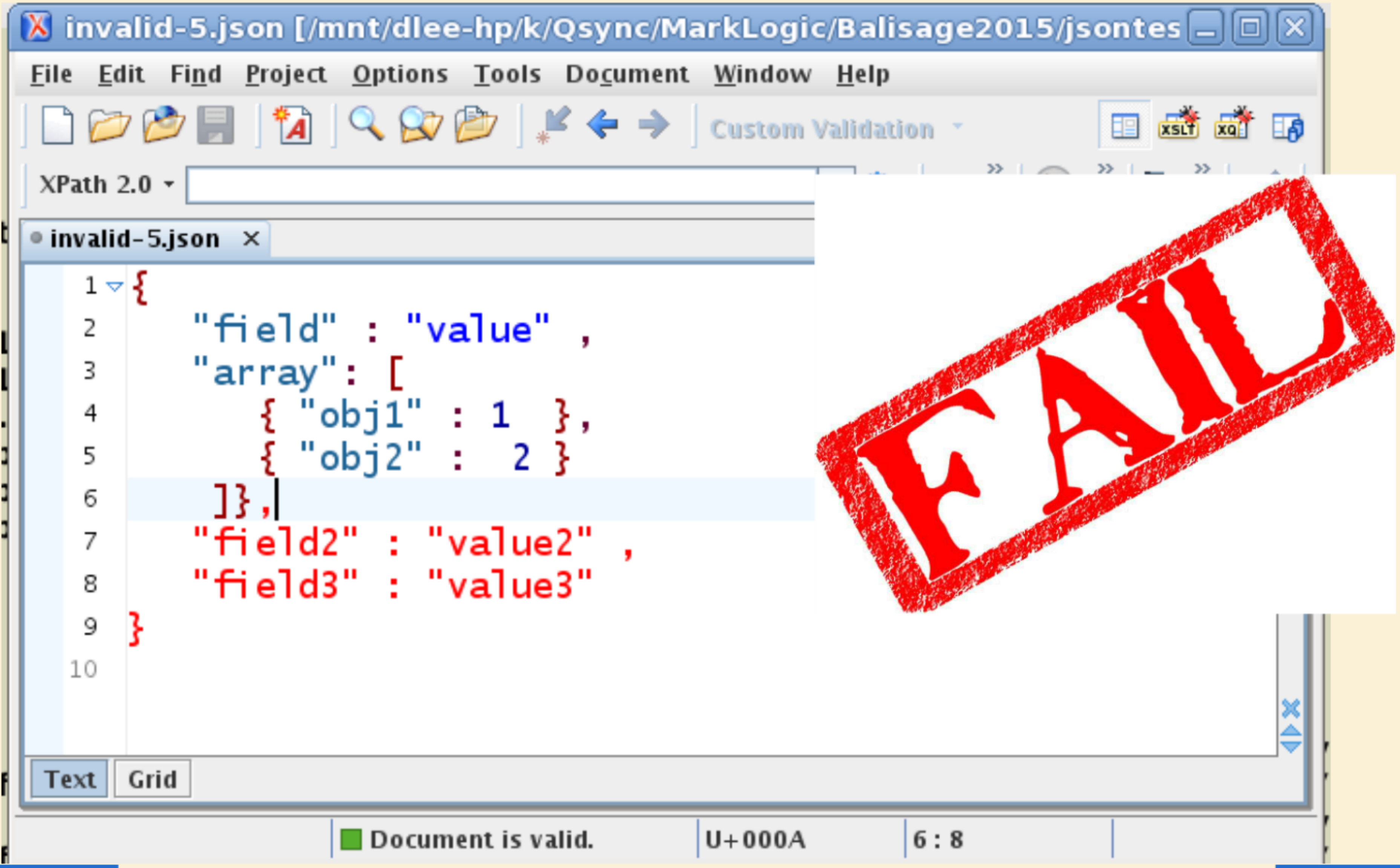
Just convert to XML

If only there were ...

A supported implementation ...
A small fixed set of elements and types ..
A Schema -- of any sort
An API that doesn't change
Formal Semantics and validation
Your software is perfect ever time
Tools and People to help ...

Just Use JSON ..

The "Simple" things are the hardest

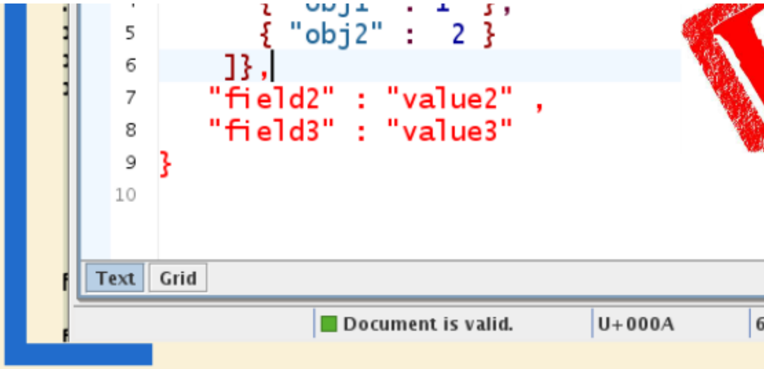


invalid-5.json [/mnt/dlee-hp/k/Qsync/MarkLogic/Balisage2015/jsontes

```
{
    "field" : "value" ,
    "array": [
       { "obj1" : 1  },
       { "obj2" :  2 }
   ]},
    "field2" : "value2" ,
    "field3" : "value3"
}
```

🟩 Document is valid. U+000A 6 : 8

t change
d validation
ct ever time
to help ...



```
      { "obj1" : 1 },
      { "obj2" :  2 }
    ]},
  "field2" : "value2" ,
  "field3" : "value3"
}
```

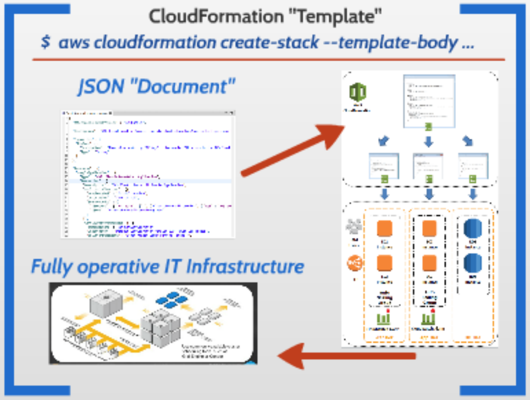
🟩 Document is valid. U+000A

... If the real cow was spherical

A Simple Document..

AWS™ Cloud Formation Templates

(tm) "Amazon Web Services"



CloudFormation "Template"

```
$ aws cloudformation create-stack --template-body ...
```

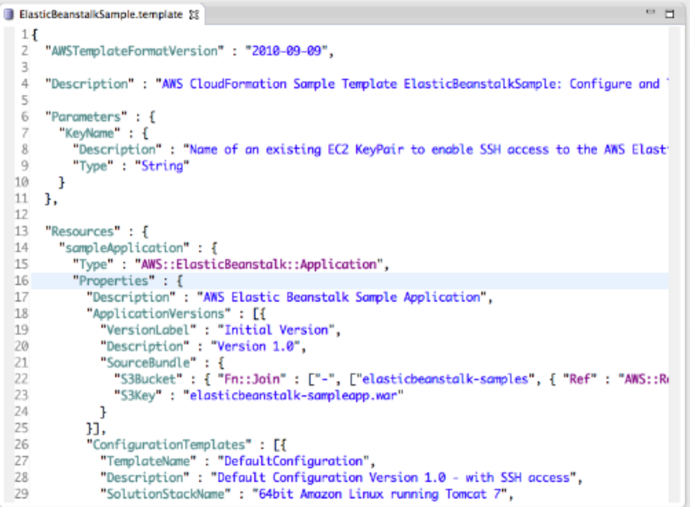
JSON "Document"

Fully operative IT Infrastructure

CloudFormation "Template"

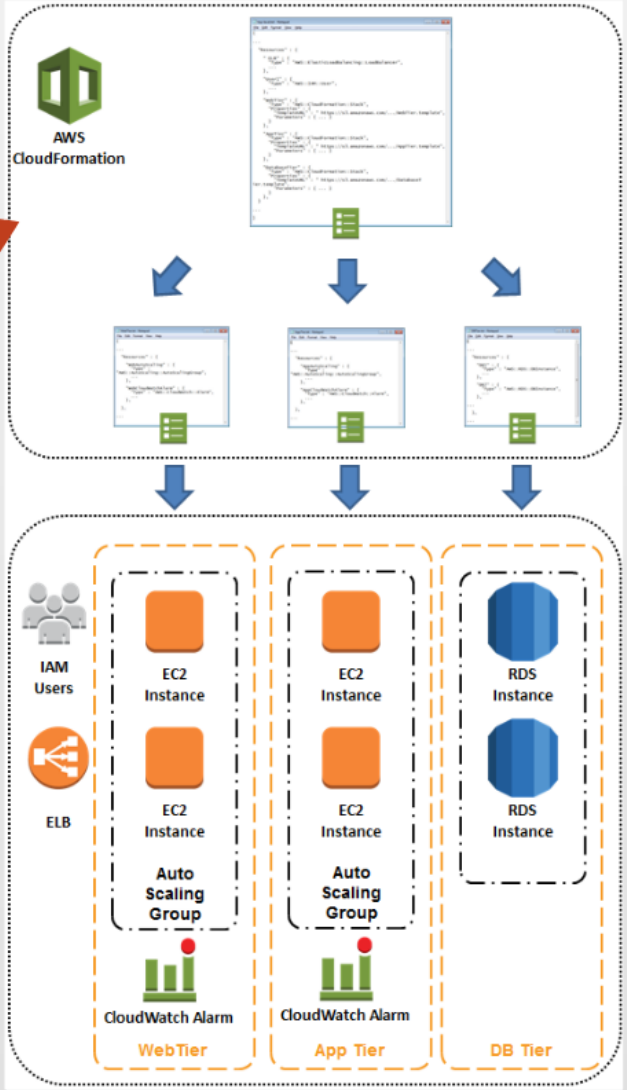
$ aws cloudformation create-stack --template-body ...

JSON "Document"

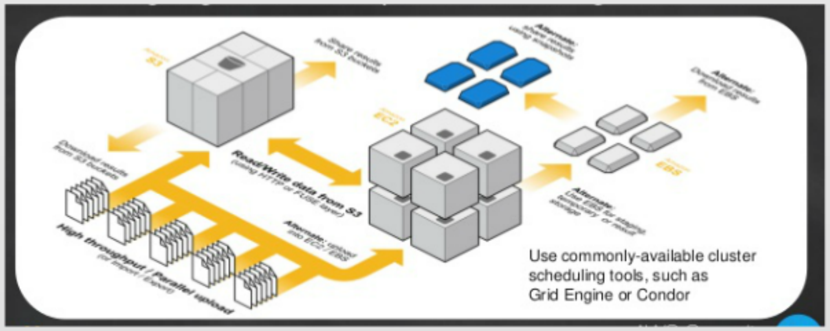


ElasticBeanstalkSample.template

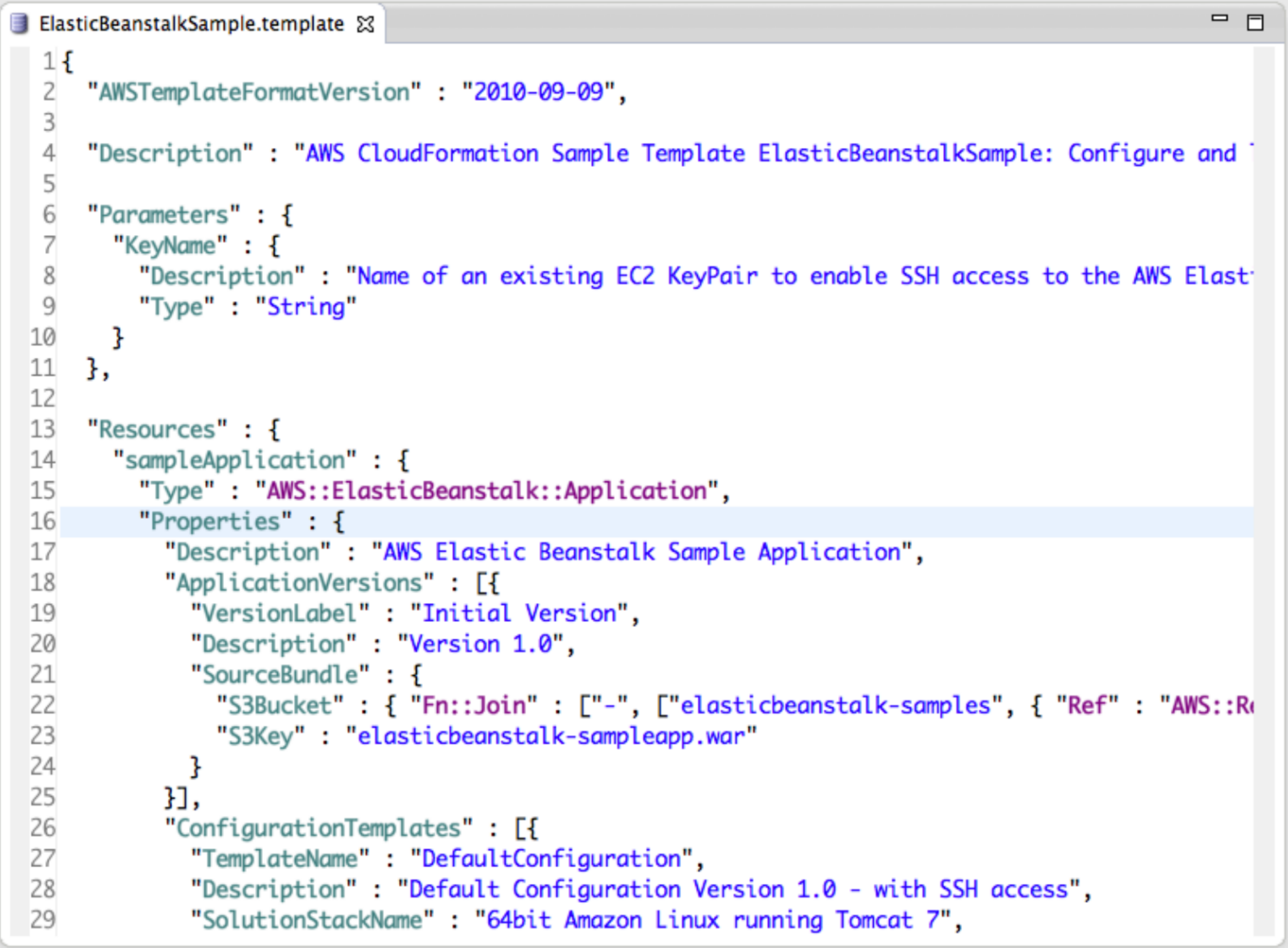
```
{
  "AWSTemplateFormatVersion" : "2010-09-09",

  "Description" : "AWS CloudFormation Sample Template ElasticBeanstalkSample: Configure and '

  "Parameters" : {
    "KeyName" : {
      "Description" : "Name of an existing EC2 KeyPair to enable SSH access to the AWS Elast
      "Type" : "String"
    }
  },

  "Resources" : {
    "sampleApplication" : {
      "Type" : "AWS::ElasticBeanstalk::Application",
      "Properties" : {
        "Description" : "AWS Elastic Beanstalk Sample Application",
        "ApplicationVersions" : [{
          "VersionLabel" : "Initial Version",
          "Description" : "Version 1.0",
          "SourceBundle" : {
            "S3Bucket" : { "Fn::Join" : ["-", ["elasticbeanstalk-samples", { "Ref" : "AWS::R
            "S3Key" : "elasticbeanstalk-sampleapp.war"
          }
        }],
        "ConfigurationTemplates" : [{
          "TemplateName" : "DefaultConfiguration",
          "Description" : "Default Configuration Version 1.0 - with SSH access",
          "SolutionStackName" : "64bit Amazon Linux running Tomcat 7",
```



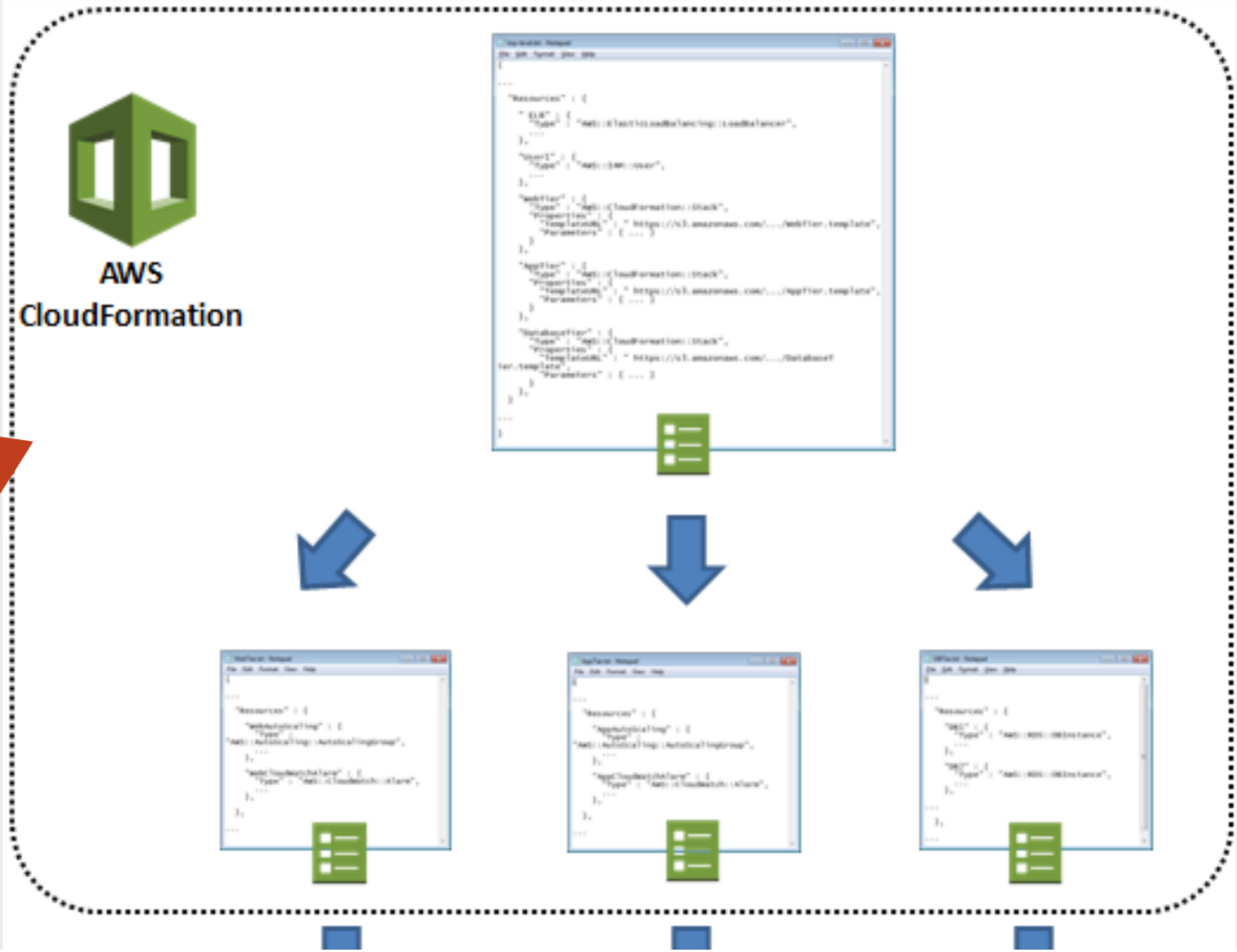
Fully operative IT Infrastructure



Use commonly-available cluster scheduling tools, such as Grid Engine or Condor



```
{
  "AWSTemplateFormatVersion" : "2010-09-09",

  "Description" : "AWS CloudFormation Sample Template ElasticBeanstalkSample: Configure and

  "Parameters" : {
    "KeyName" : {
      "Description" : "Name of an existing EC2 KeyPair to enable SSH access to the AWS Elast
      "Type" : "String"
    }
  },

  "Resources" : {
    "sampleApplication" : {
      "Type" : "AWS::ElasticBeanstalk::Application",
      "Properties" : {
        "Description" : "AWS Elastic Beanstalk Sample Application",
        "ApplicationVersions" : [{
          "VersionLabel" : "Initial Version",
          "Description" : "Version 1.0",
          "SourceBundle" : {
            "S3Bucket" : { "Fn::Join" : ["-", ["elasticbeanstalk-samples", { "Ref" : "AWS::R
            "S3Key" : "elasticbeanstalk-sampleapp.war"
          }
        }],
        "ConfigurationTemplates" : [{
          "TemplateName" : "DefaultConfiguration",
          "Description" : "Default Configuration Version 1.0 - with SSH access",
          "SolutionStackName" : "64bit Amazon Linux running Tomcat 7",
```

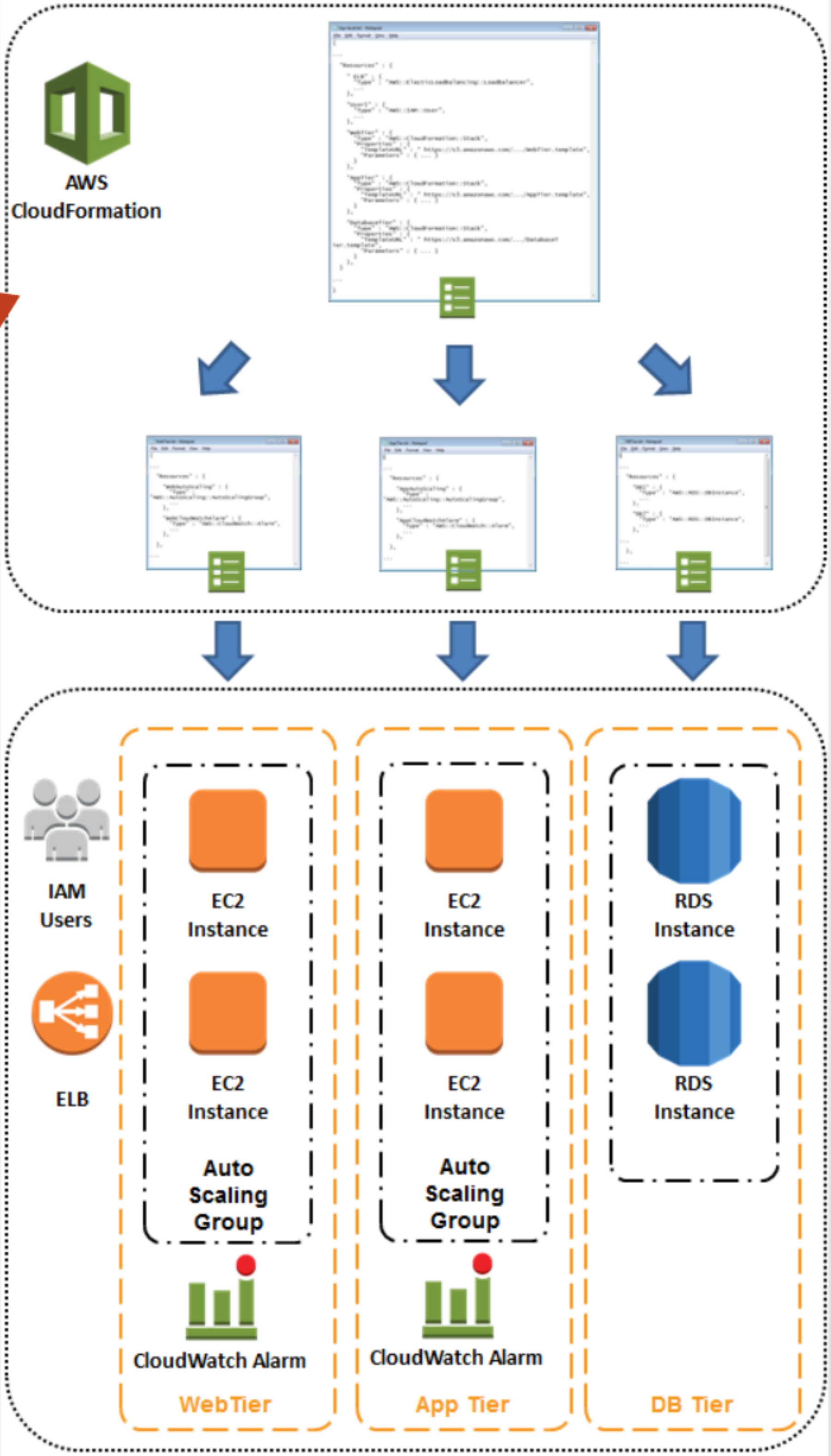


```
{
...
  "Resources" : {
    " ELB" : {
      "Type" : "AWS::ElasticLoadBalancing::LoadBalancer",
      ...
    },
    "User1" : {
      "Type" : "AWS::IAM::User",
      ...
    },
    "WebTier" : {
      "Type" : "AWS::CloudFormation::Stack",
      "Properties" : {
        "TemplateURL" : " https://s3.amazonaws.com/.../WebTier.template",
        "Parameters" : { ... }
      }
    },
    "AppTier" : {
      "Type" : "AWS::CloudFormation::Stack",
      "Properties" : {
        "TemplateURL" : " https://s3.amazonaws.com/.../AppTier.template",
        "Parameters" : { ... }
      }
    },
    "DatabaseTier" : {
      "Type" : "AWS::CloudFormation::Stack",
      "Properties" : {
        "TemplateURL" : " https://s3.amazonaws.com/.../DatabaseT
ier.template",
        "Parameters" : { ... }
      }
    },
  }
...
}
```

```
{
...
  "Resources" : {
    "WebAutoScaling" : {
      "Type" :
"AWS::AutoScaling::AutoScalingGroup",
      ...
    },
    "WebCloudWatchAlarm" : {
      "Type" : "AWS::CloudWatch::Alarm",
      ...
    },
  },
...
```

```
{
...
  "Resources" : {
    "AppAutoScaling" : {
      "Type" :
"AWS::AutoScaling::AutoScalingGroup",
      ...
    },
    "AppCloudWatchAlarm" : {
      "Type" : "AWS::CloudWatch::Alarm",
      ...
    },
  },
...
```

```
{
...
  "Resources" : {
    "DB1" : {
      "Type" : "AWS::RDS::DBInstance",
      ...
    },
    "DB2" : {
      "Type" : "AWS::RDS::DBInstance",
      ...
    },
...
  },
...
```



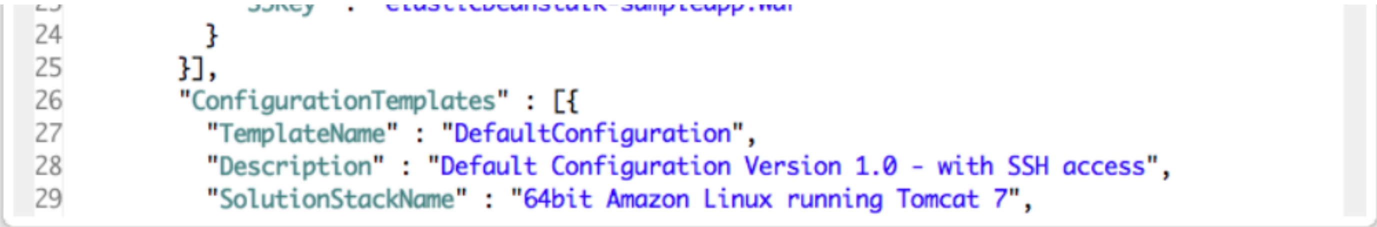
```
{
...
  "Resources" : {
    " ELB" : {
      "Type" : "AWS::ElasticLoadBalancing::LoadBalancer",
      ...
    },
    "User1" : {
      "Type" : "AWS::IAM::User",
      ...
    },
    "WebTier" : {
      "Type" : "AWS::CloudFormation::Stack",
      "Properties" : {
        "TemplateURL" : " https://s3.amazonaws.com/.../WebTier.template",
        "Parameters" : { ... }
      }
    },
    "AppTier" : {
      "Type" : "AWS::CloudFormation::Stack",
      "Properties" : {
        "TemplateURL" : " https://s3.amazonaws.com/.../AppTier.template",
        "Parameters" : { ... }
      }
    },
    "DatabaseTier" : {
      "Type" : "AWS::CloudFormation::Stack",
      "Properties" : {
        "TemplateURL" : " https://s3.amazonaws.com/.../DatabaseT
ier.template",
        "Parameters" : { ... }
      }
    },
  }
...
}
```

```
{
...
  "Resources" : {
    "WebAutoScaling" : {
      "Type" :
"AWS::AutoScaling::AutoScalingGroup",
      ...
    },
    "WebCloudWatchAlarm" : {
      "Type" : "AWS::CloudWatch::Alarm",
      ...
    },
  },
...
```

```
{
...
  "Resources" : {
    "AppAutoScaling" : {
      "Type" :
"AWS::AutoScaling::AutoScalingGroup",
      ...
    },
    "AppCloudWatchAlarm" : {
      "Type" : "AWS::CloudWatch::Alarm",
      ...
    },
  },
...
```

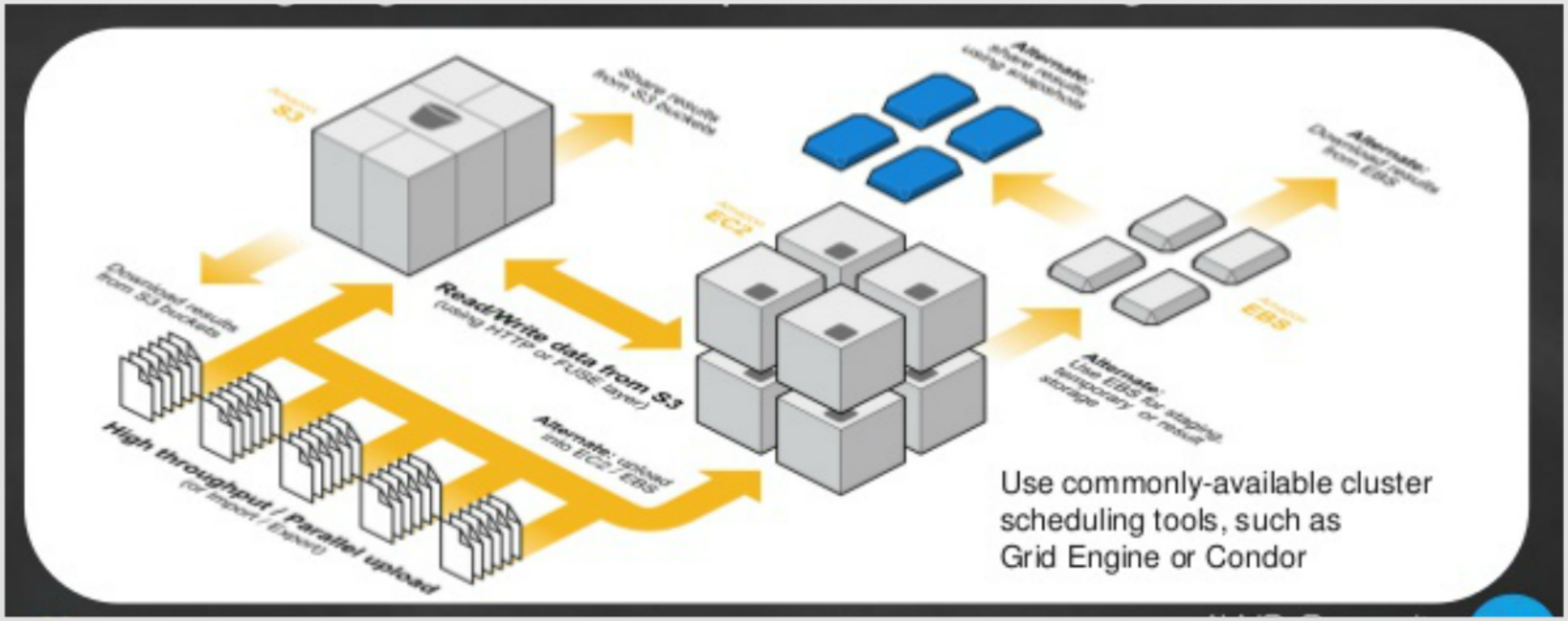
```
{
...
  "Resources" : {
    "DB1" : {
      "Type" : "AWS::RDS::DBInstance",
      ...
    },
    "DB2" : {
      "Type" : "AWS::RDS::DBInstance",
      ...
    },
...
  },
...
```

tructure



```
            "S3Key" : "elasticbeanstalk-sampleapp.war"
          }
        }],
        "ConfigurationTemplates" : [{
          "TemplateName" : "DefaultConfiguration",
          "Description" : "Default Configuration Version 1.0 - with SSH access",
          "SolutionStackName" : "64bit Amazon Linux running Tomcat 7",
```

Fully operative IT Infrastructure


Amazon S3
Share results from S3 buckets
Alternate: share results using snapshots
Amazon EC2
Alternate: Download results from EBS
Amazon EBS
Download results from S3 buckets
Read/Write data from S3 (using HTTP or FUSE layer)
Alternate: Use EBS for staging temporary or result storage
Alternate: upload into EC2 / EBS
High throughput / Parallel upload (or Import / Export)

Use commonly-available cluster scheduling tools, such as Grid Engine or Condor

Document Structure

Structured
Semi Structured
Unstructured

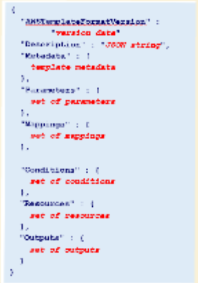
Root Structure

Not bad at all ...

- *Simple*
- *Fixed 'schema' (*)*
- *Compatible names/types*

(*) "schema"

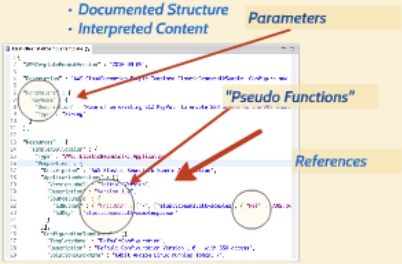
- No machine readable
- Good prose documentation
- Accuracy varies



```
{
  "AWSTemplateFormatVersion" :
        "version date"
  "Description" : "JSON string",
  "Metadata" : {
    template metadata
  },
  "Parameters" : {
    set of parameters
  },
  "Mappings" : {
    set of mappings
  },

  "Conditions" : {
    set of conditions
  },
  "Resources" : {
    set of resources
  },
  "Outputs" : {
    set of outputs
  }
}
```

First Nested Structure

- *~100 2nd level types*
- *Documented Structure*
- *Interpreted Content*



And then it gets a little worse ...

Deployment / orchestration / OS command scripting
Embedded binaries / installed and executed
You don't want to know what this is ...

Document Structure

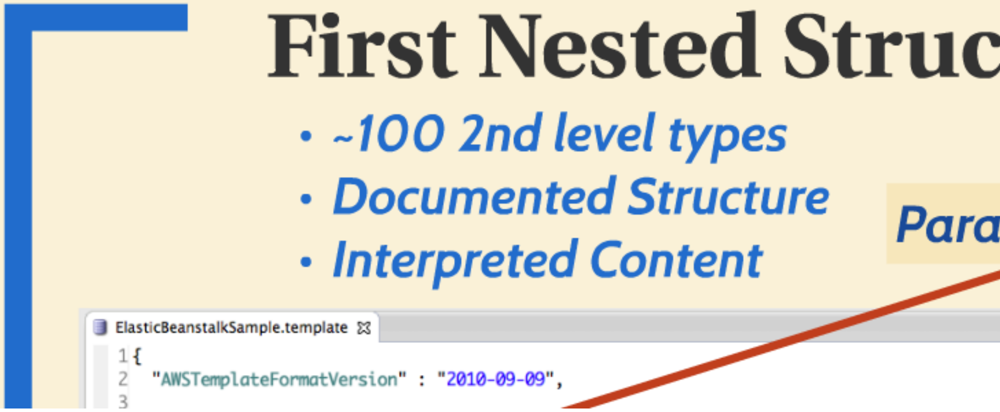
Structured

Semi Structured

Unstructured

ucture

```
{
  "AWSTemplateFormatVersion" :
        "version date"
```



First Nested Struc

- *~100 2nd level types*
- *Documented Structure*
- *Interpreted Content*

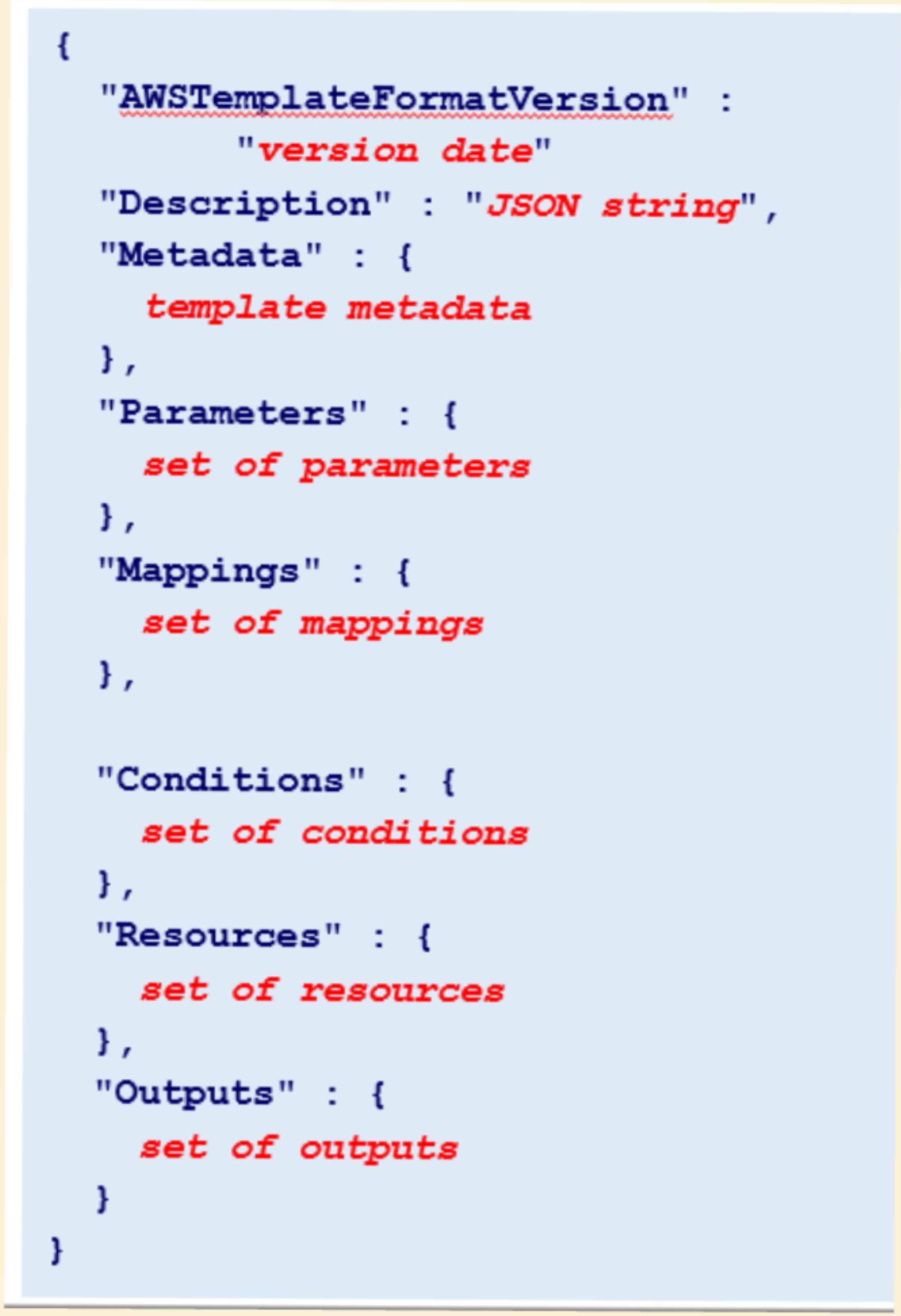
Para

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
```

Root Structure

Not bad at all ...

- *Simple*
- *Fixed 'schema' (*)*
- *Compatible names/types*



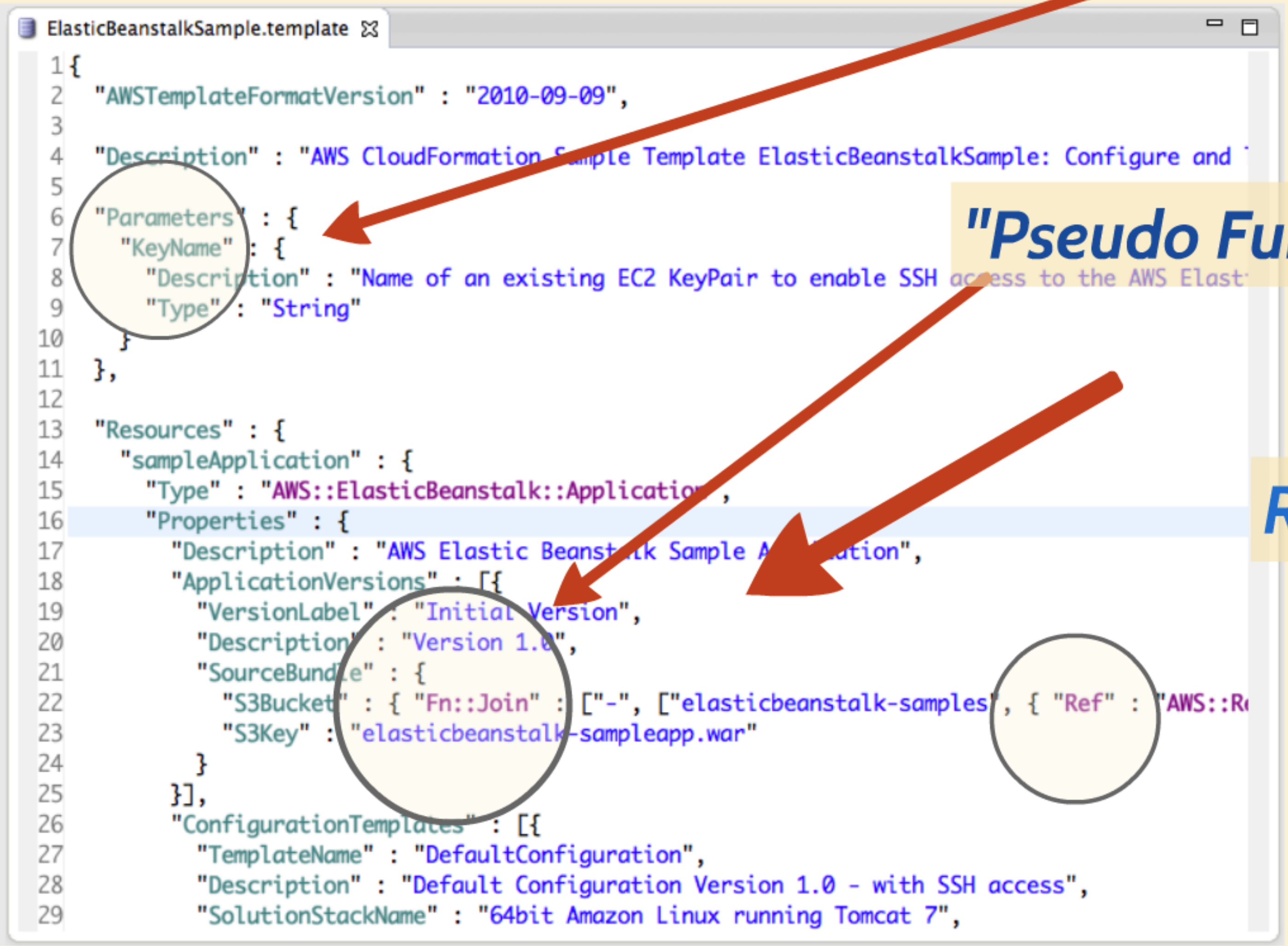
```
{
  "AWSTemplateFormatVersion" :
        "version date"
  "Description" : "JSON string",
  "Metadata" : {
    template metadata
  },
  "Parameters" : {
    set of parameters
  },
  "Mappings" : {
    set of mappings
  },

  "Conditions" : {
    set of conditions
  },
  "Resources" : {
    set of resources
  },
  "Outputs" : {
    set of outputs
  }
}
```

(*) "schema"

- No machine readable
- Good prose documentation
- Accuracy varies

First Nested Structure

- *~100 2nd level types*
- *Documented Structure*
- *Interpreted Content*



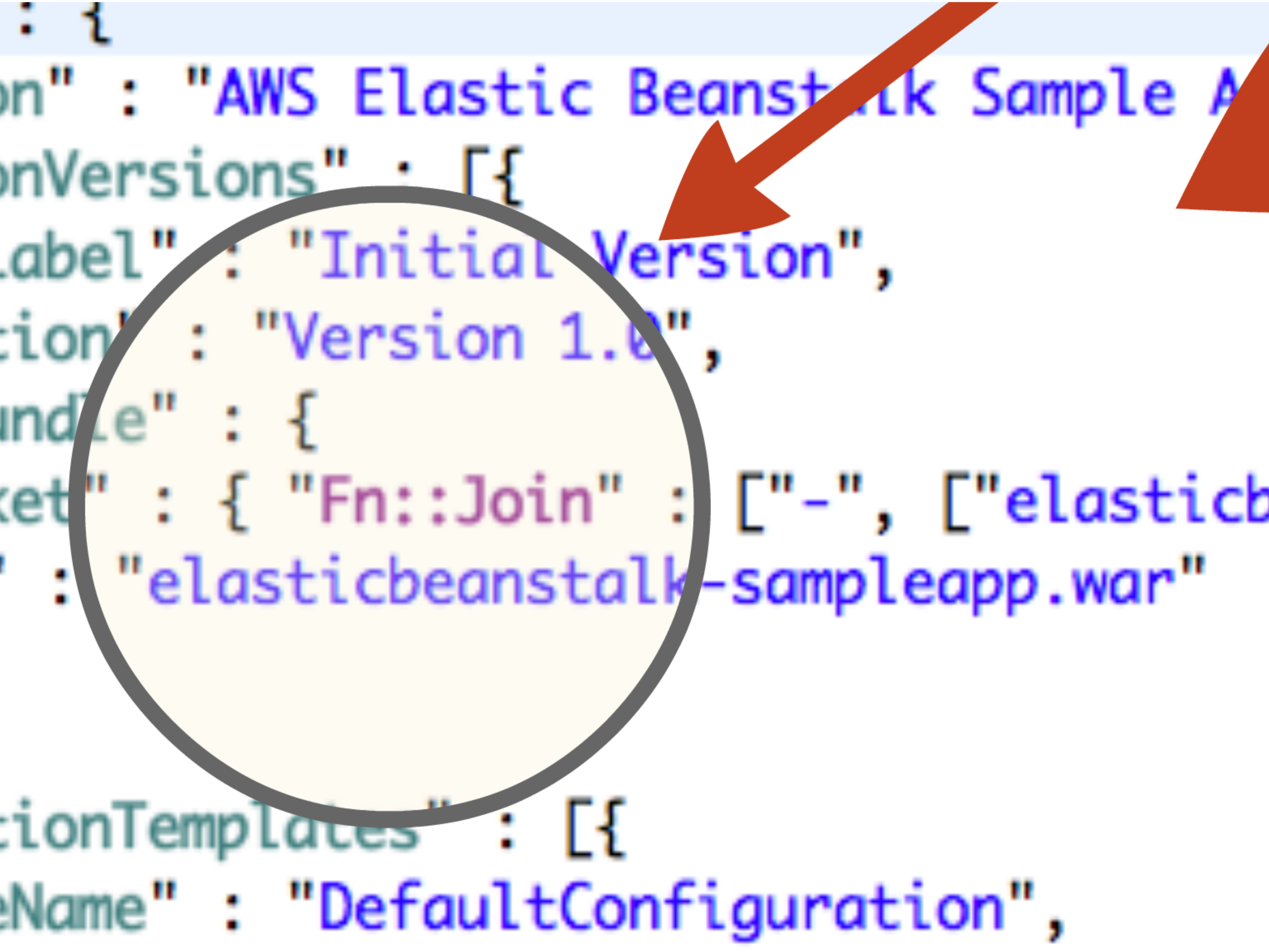
ElasticBeanstalkSample.template

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",

  "Description" : "AWS CloudFormation Sample Template ElasticBeanstalkSample: Configure and 

  "Parameters" : {
    "KeyName" : {
      "Description" : "Name of an existing EC2 KeyPair to enable SSH access to the AWS Elast
      "Type" : "String"
    }
  },

  "Resources" : {
    "sampleApplication" : {
      "Type" : "AWS::ElasticBeanstalk::Application",
      "Properties" : {
        "Description" : "AWS Elastic Beanstalk Sample Application",
        "ApplicationVersions" : [{
          "VersionLabel" : "Initial Version",
          "Description" : "Version 1.0",
          "SourceBundle" : {
            "S3Bucket" : { "Fn::Join" : ["-", ["elasticbeanstalk-samples", { "Ref" : "AWS::Re
            "S3Key" : "elasticbeanstalk-sampleapp.war"
          }
        }],
        "ConfigurationTemplates" : [{
          "TemplateName" : "DefaultConfiguration",
          "Description" : "Default Configuration Version 1.0 - with SSH access",
          "SolutionStackName" : "64bit Amazon Linux running Tomcat 7",
```

Parameters

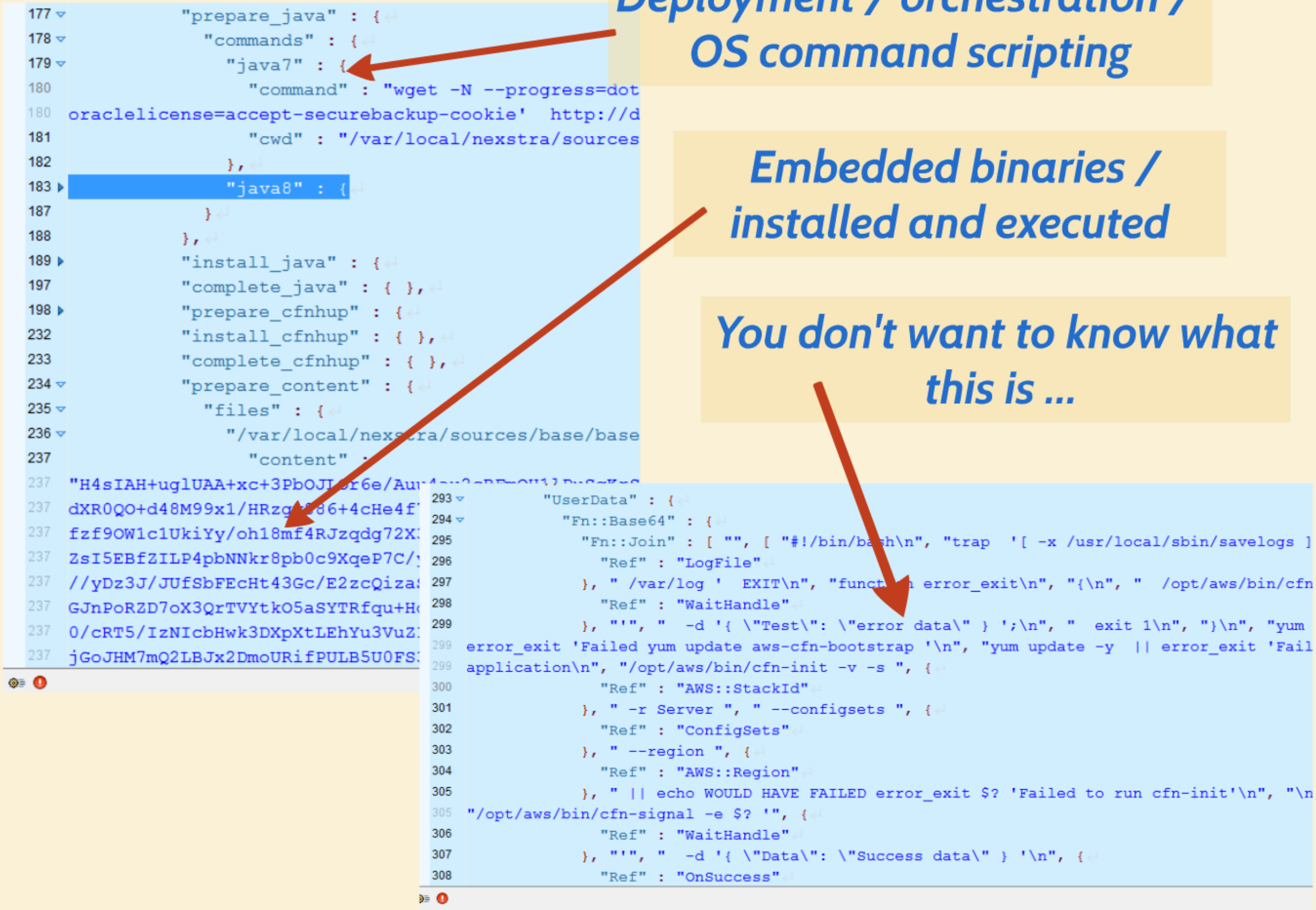
"Pseudo Functions"

References



```
 : {
on" : "AWS Elastic Beanstalk Sample A
onVersions" : [{
Label" : "Initial Version",
tion" : "Version 1.0",
undle" : {
ket" : { "Fn::Join" : ["-", ["elasticb
' : "elasticbeanstalk-sampleapp.war"


tionTemplates" : [{
eName" : "DefaultConfiguration",
```

And then it gets a little worse ...



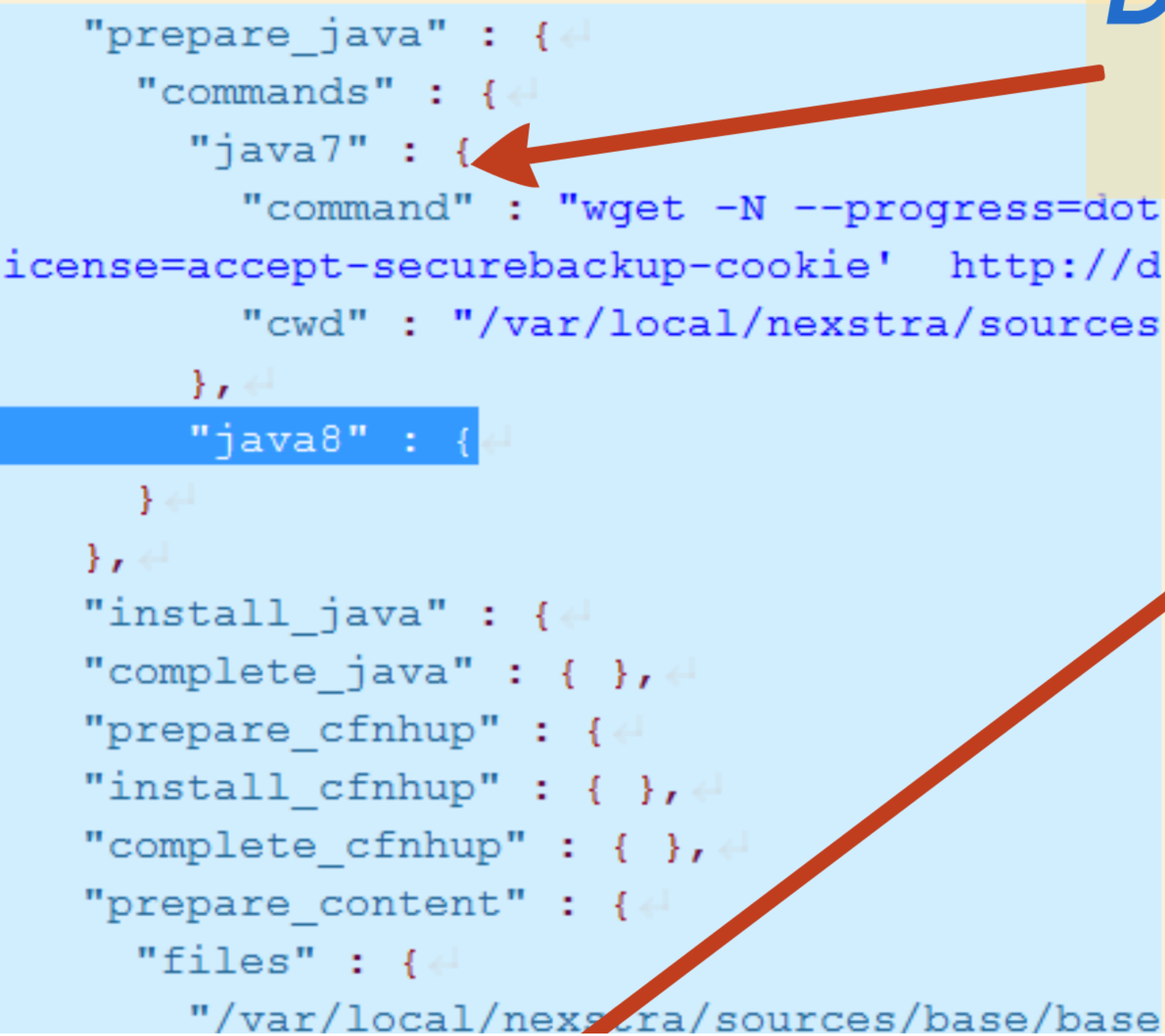
```
          "prepare_java" : {
            "commands" : {
              "java7" : {
                "command" : "wget -N --progress=dot
oraclelicense=accept-securebackup-cookie'  http://d
                "cwd" : "/var/local/nexstra/sources
              },
              "java8" : {
            }
          },
          "install_java" : {
          "complete_java" : { },
          "prepare_cfnhup" : {
          "install_cfnhup" : { },
          "complete_cfnhup" : { },
          "prepare_content" : {
            "files" : {
              "/var/local/nexstra/sources/base/base
                "content" :
"H4sIAH+uglUAA+xc+3PbOJLOr6e/Au
dXR0QO+d48M99x1/HRzg[illegible]86+4cHe4f
fzf9OW1c1UkiYy/oh18mf4RJzqdg72X
ZsI5EBfZILP4pbNNkr8pb0c9XqeP7C/
//yDz3J/JUfSbFEcHt43Gc/E2zcQiza
GJnPoRZD7oX3QrTVYtkO5aSYTRfqu+H
0/cRT5/IzNIcbHwk3DXpXtLEhYu3VuZ
jGoJHM7mQ2LBJx2DmoURifPULB5U0FS
```

```
          "UserData" : {
            "Fn::Base64" : {
              "Fn::Join" : [ "", [ "#!/bin/bash\n", "trap  '[ -x /usr/local/sbin/savelogs ]
                "Ref" : "LogFile"
              }, " /var/log '  EXIT\n", "function error_exit\n", "{\n", "  /opt/aws/bin/cfn
                "Ref" : "WaitHandle"
              }, "'", "  -d '{ \"Test\": \"error data\" } ';\n", "  exit 1\n", "}\n", "yum
error_exit 'Failed yum update aws-cfn-bootstrap '\n", "yum update -y  || error_exit 'Fail
application\n", "/opt/aws/bin/cfn-init -v -s ", {
                "Ref" : "AWS::StackId"
              }, " -r Server ", " --configsets ", {
                "Ref" : "ConfigSets"
              }, " --region ", {
                "Ref" : "AWS::Region"
              }, " || echo WOULD HAVE FAILED error_exit $? 'Failed to run cfn-init'\n", "\n
"/opt/aws/bin/cfn-signal -e $? '", {
                "Ref" : "WaitHandle"
              }, "'", "  -d '{ \"Data\": \"Success data\" } '\n", {
                "Ref" : "OnSuccess"
```

Deployment / orchestration / OS command scripting

Embedded binaries / installed and executed

You don't want to know what this is ...



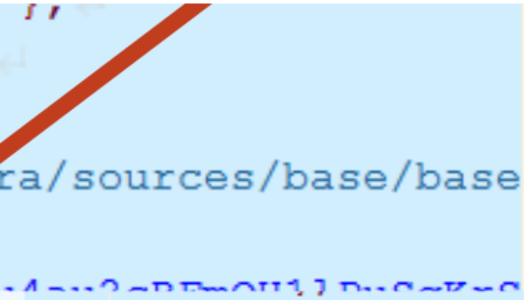
```
    "prepare_java" : {↵
      "commands" : {↵
        "java7" : {↵
          "command" : "wget -N --progress=dot
icense=accept-securebackup-cookie'  http://d
          "cwd" : "/var/local/nexstra/sources
        },↵
        "java8" : {↵
      }↵
    },↵
    "install_java" : {↵
    "complete_java" : { },↵
    "prepare_cfnhup" : {↵
    "install_cfnhup" : { },↵
    "complete_cfnhup" : { },↵
    "prepare_content" : {↵
      "files" : {↵
        "/var/local/nexstra/sources/base/base
```

Deploym
OS co

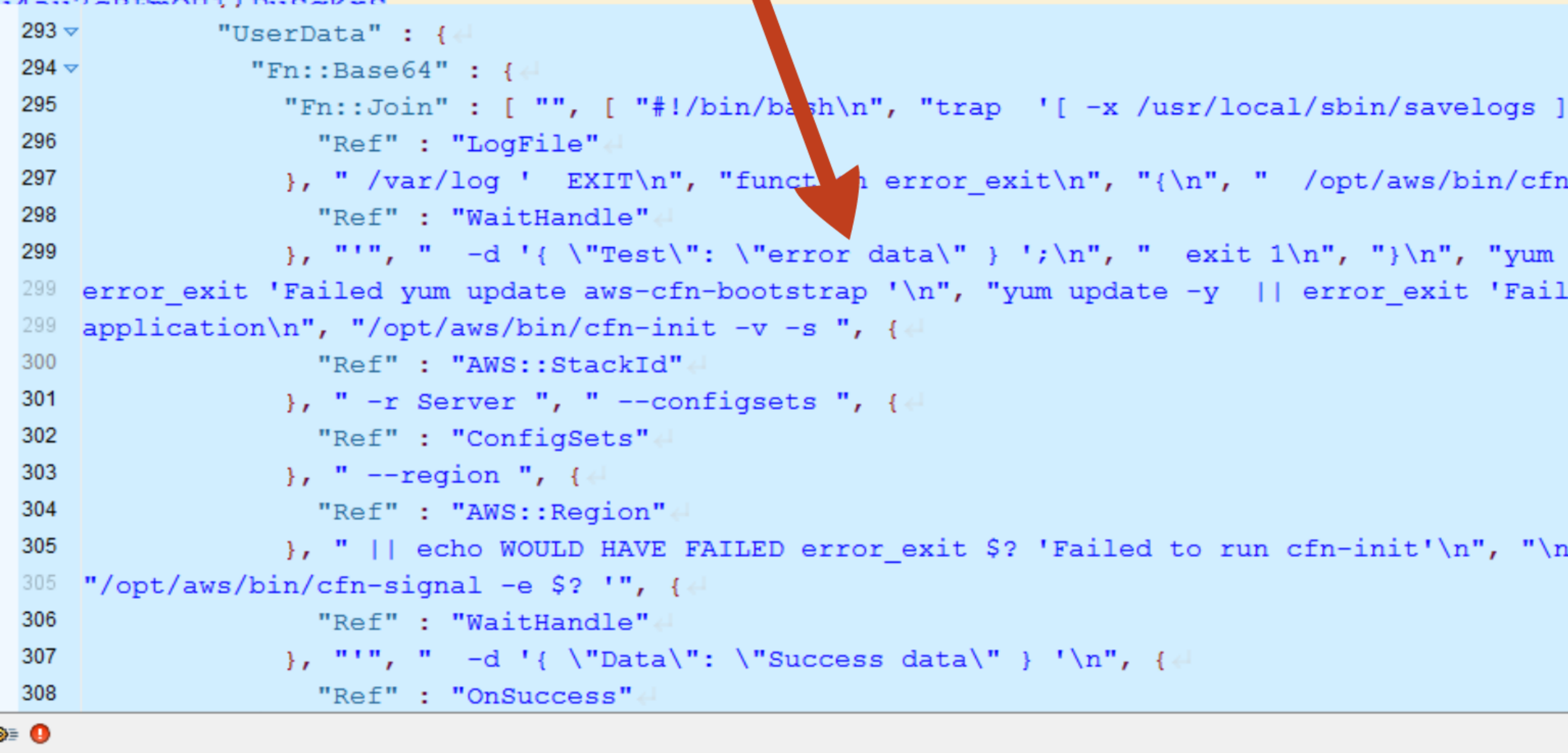
En
ins

You

this is ...

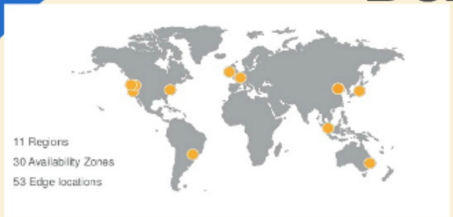


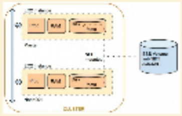
```
ra/sources/base/base
```

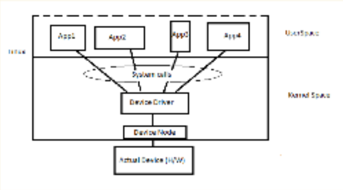


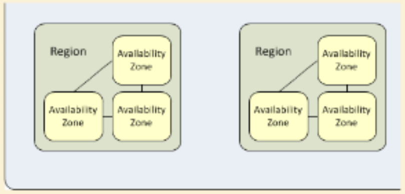
```
          "UserData" : {
            "Fn::Base64" : {
              "Fn::Join" : [ "", [ "#!/bin/bash\n", "trap  '[ -x /usr/local/sbin/savelogs ]
                "Ref" : "LogFile"
              }, " /var/log '  EXIT\n", "function error_exit\n", "{\n", "  /opt/aws/bin/cfn
                "Ref" : "WaitHandle"
              }, "'", "  -d '{ \"Test\": \"error data\" } ';\n", "  exit 1\n", "}\n", "yum 
error_exit 'Failed yum update aws-cfn-bootstrap '\n", "yum update -y  || error_exit 'Fail
application\n", "/opt/aws/bin/cfn-init -v -s ", {
                "Ref" : "AWS::StackId"
              }, " -r Server ", " --configsets ", {
                "Ref" : "ConfigSets"
              }, " --region ", {
                "Ref" : "AWS::Region"
              }, " || echo WOULD HAVE FAILED error_exit $? 'Failed to run cfn-init'\n", "\n
"/opt/aws/bin/cfn-signal -e $? '", {
                "Ref" : "WaitHandle"
              }, "'", "  -d '{ \"Data\": \"Success data\" } '\n", {
                "Ref" : "OnSuccess"
```

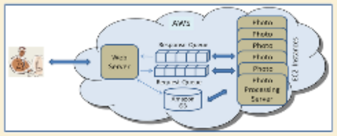
Scale & Scope of the Domain

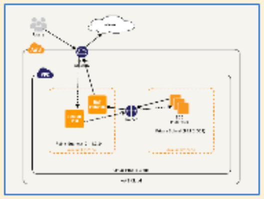


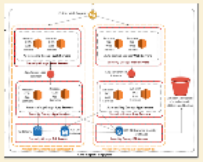


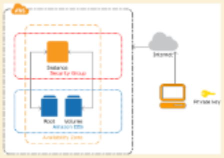




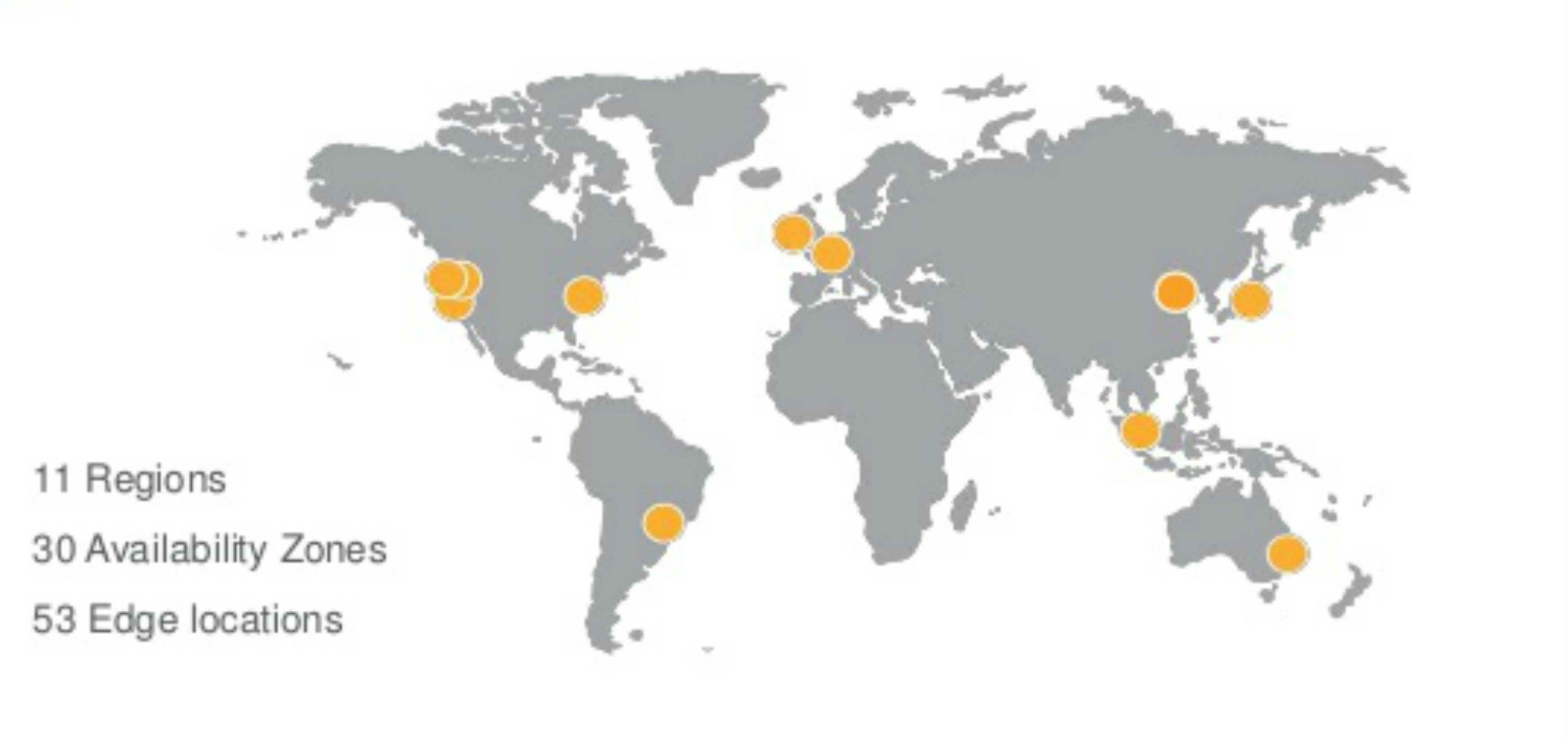








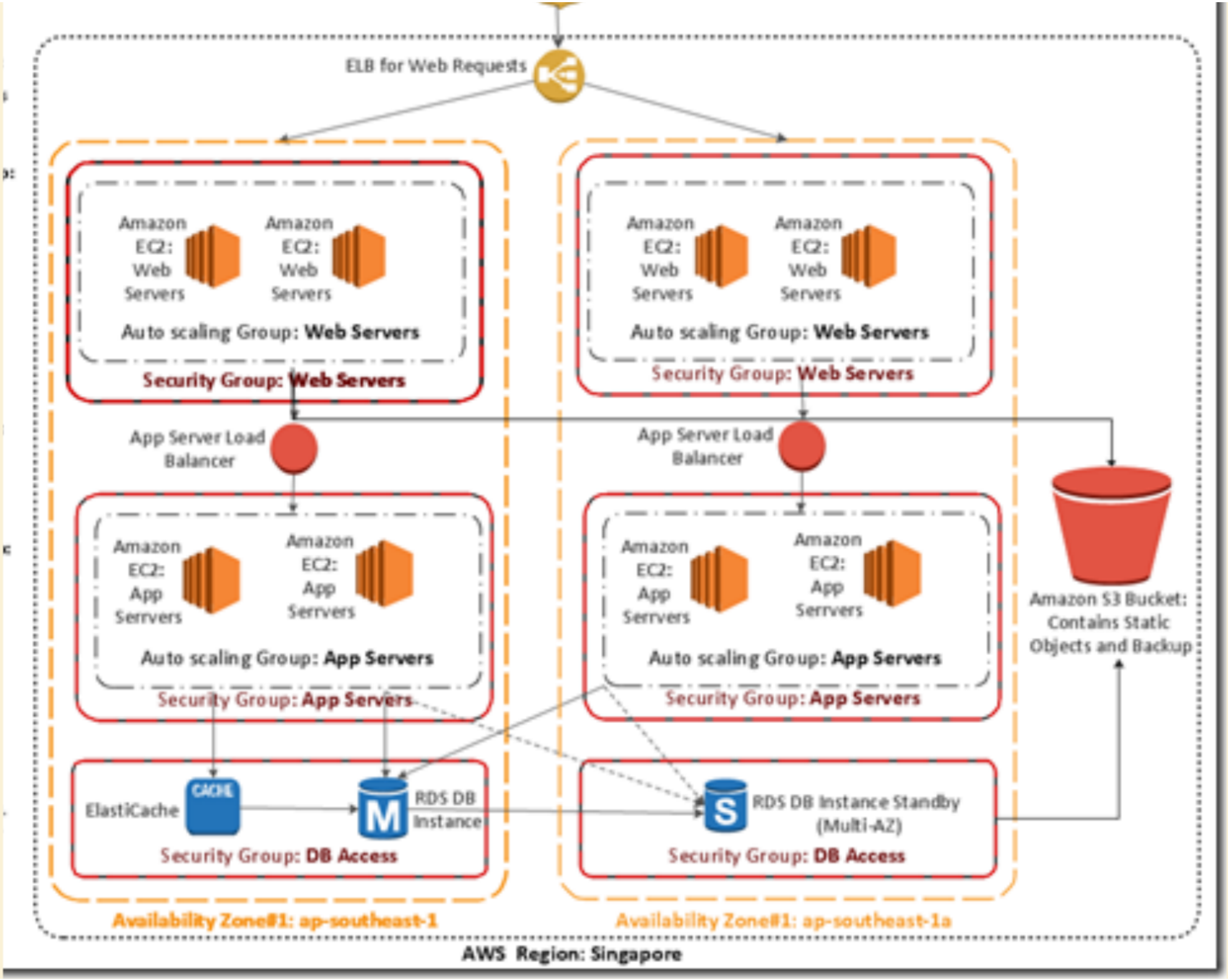
Do

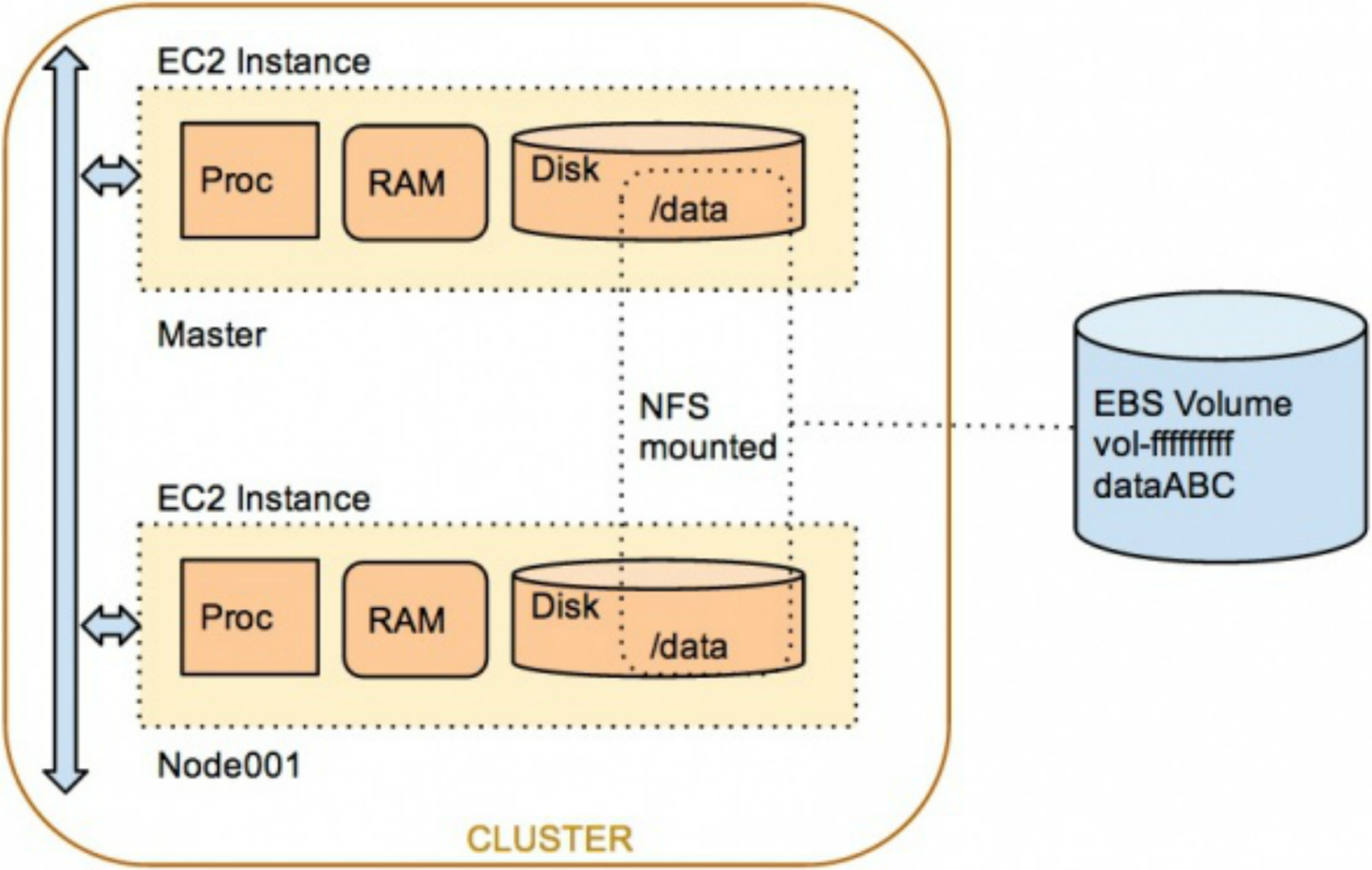


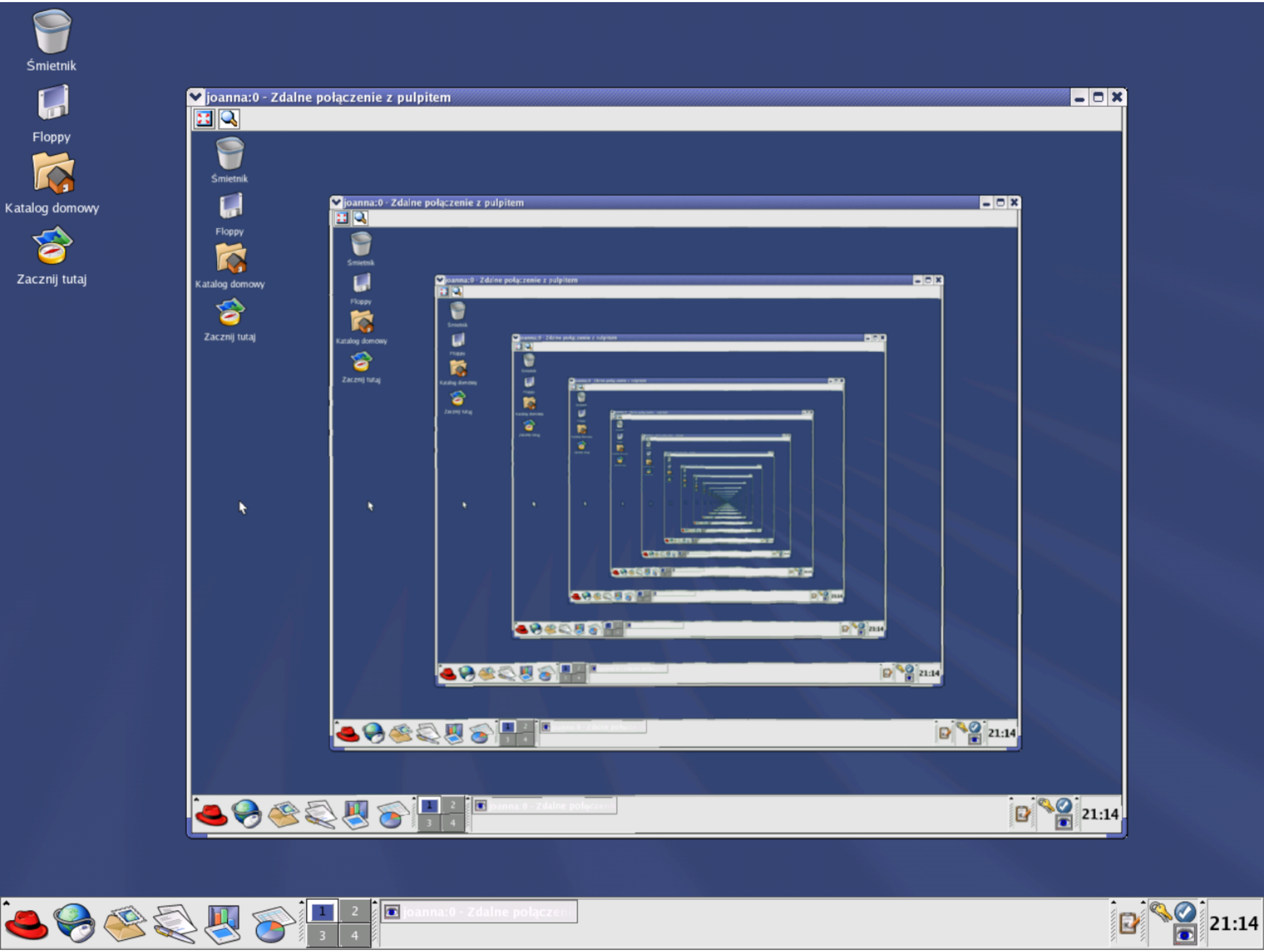
11 Regions

30 Availability Zones

53 Edge locations


ELB for Web Requests
Amazon EC2: Web Servers
Amazon EC2: Web Servers
Auto scaling Group: Web Servers
Security Group: Web Servers
Amazon EC2: Web Servers
Amazon EC2: Web Servers
Auto scaling Group: Web Servers
Security Group: Web Servers
App Server Load Balancer
App Server Load Balancer
Amazon EC2: App Serrvers
Amazon EC2: App Serrvers
Auto scaling Group: App Servers
Security Group: App Servers
Amazon EC2: App Serrvers
Amazon EC2: App Serrvers
Auto scaling Group: App Servers
Security Group: App Servers
Amazon S3 Bucket: Contains Static Objects and Backup
ElastiCache
CACHE
M
RDS DB Instance
Security Group: DB Access
S
RDS DB Instance Standby (Multi-AZ)
Security Group: DB Access
Availability Zone#1: ap-southeast-1
Availability Zone#1: ap-southeast-1a
AWS Region: Singapore


EC2 Instance
Proc
RAM
Disk
/data
Master
NFS
mounted
EBS Volume
vol-fffffffff
dataABC
EC2 Instance
Proc
RAM
Disk
/data
Node001
CLUSTER


Śmietnik
Floppy
Katalog domowy
Zacznij tutaj
joanna:0 - Zdalne połączenie z pulpitem
1
2
3
4
joanna:0 - Zdalne połączeni
21:14

#Assumption

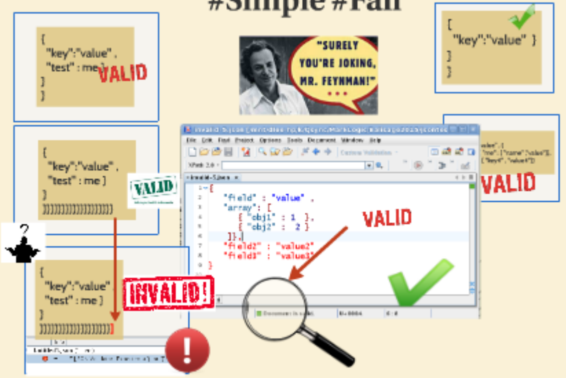
Basic Document Editing

Should obviously be simple ..

"Well Formed" Validation

#Assumption

validation of obvious errors
#Simple #Fail



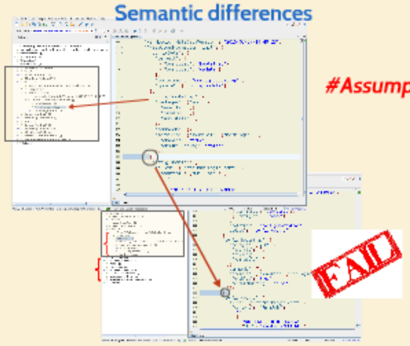
```
{
 "key":"value" ,
 "test" : me }
}
}
```

```
[
 "key":"value" }
]
}
```

```
{
 "key":"value" ,
 "test" : me }
}
}}}}}}}}}}}}}}}}}
```

```
{
 "field" : "value" ,
 "array": [
    { "obj1" : 1 },
    { "obj2" : 2 }
 ]},
 "field2" : "value2"
 "field3" : "value3"
}
```

Small format changes cause small Semantic differences



#Assumption

#Assumption

Basic Document Editing

Should obviously be simple ..

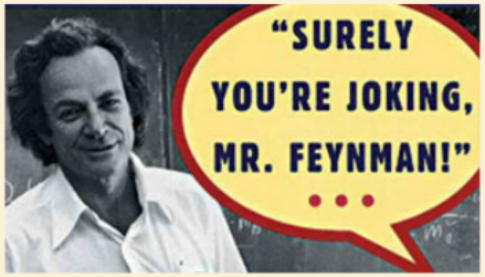
Formed" Validation #Assumption

dation of obvious errors

validation of obvious errors
#Simple #Fail

```
{
 "key":"value" ,
 "test" : me }
}
}
```

VALID



```
{
 "key":"value" }
}
}
```

✓

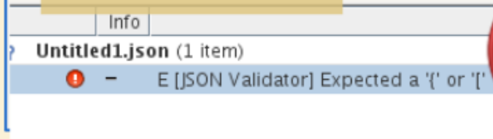
```
{
 "key":"value" ,
 "test" : me }
}
}}}}}}}}}}}}}}}}}}}}
```

VALID

Valuing Active Life in Dementia

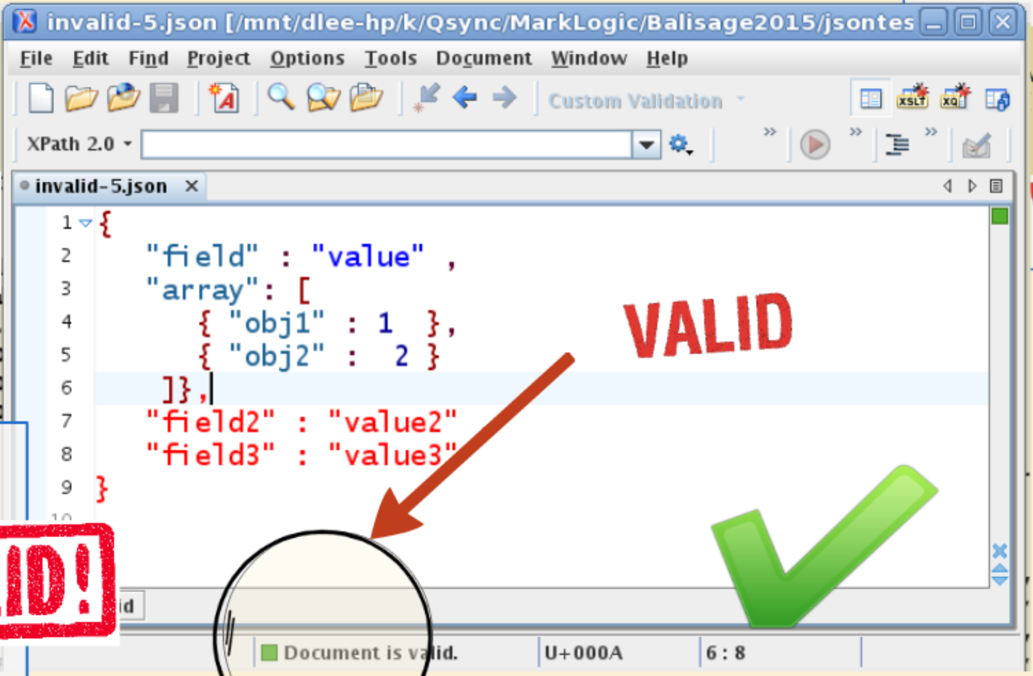
```
{
 "key":"value" ,
 "test" : me }
}
}}}}}}}}}}}}}}}}}}}]
```

INVALID!



Untitled1.json (1 item)

E [JSON Validator] Expected a '{' or '['


invalid-5.json [/mnt/dlee-hp/k/Qsync/MarkLogic/Balisage2015/jsontes
File Edit Find Project Options Tools Document Window Help
Custom Validation
XPath 2.0
invalid-5.json

```
{
    "field" : "value" ,
    "array": [
        { "obj1" : 1  },
        { "obj2" :  2 }
    ]},
    "field2" : "value2"
    "field3" : "value3"
}
```

Document is valid. U+000A 6 : 8

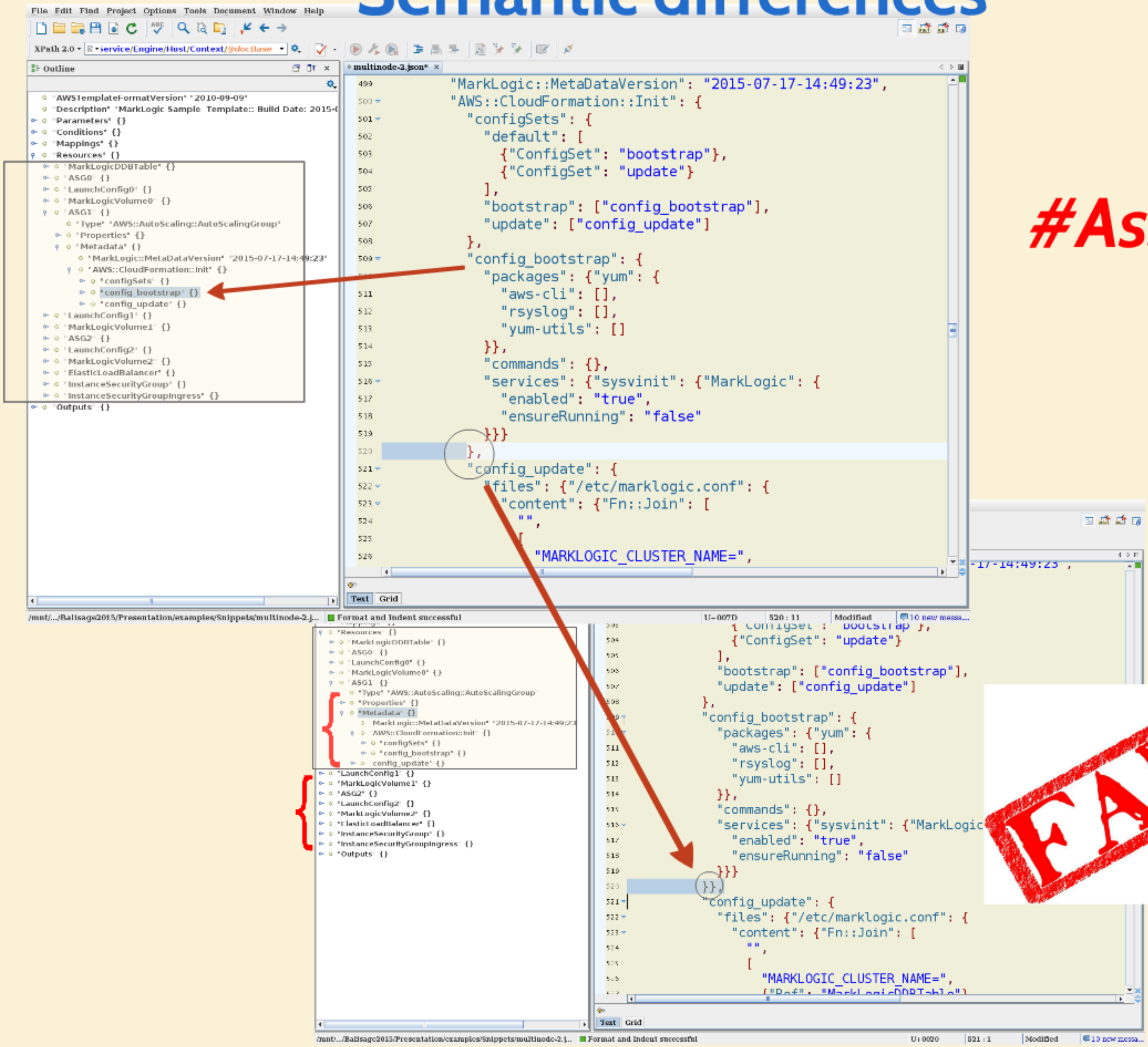
VALID

✓

```
value" , {
"me" : [ "name" ,"value"]},
[ "key4" , "value4"]}
```

VALID

Small format changes cause small Semantic differences



File Edit Find Project Options Tools Document Window Help
XPath 2.0
service/Engine/Host/Context/@docBase
Outline
multinode-2.json*
Text
Grid

```
 "AWSTemplateFormatVersion" "2010-09-09"
 "Description" "MarkLogic Sample Template:: Build Date: 2015-(
 "Parameters" {}
 "Conditions" {}
 "Mappings" {}
 "Resources" {}
   "MarkLogicDDBTable" {}
   "ASG0" {}
   "LaunchConfig0" {}
   "MarkLogicVolume0" {}
   "ASG1" {}
     "Type" "AWS::AutoScaling::AutoScalingGroup"
     "Properties" {}
     "Metadata" {}
       "MarkLogic::MetaDataVersion" "2015-07-17-14:49:23"
       "AWS::CloudFormation::Init" {}
         "configSets" {}
         "config_bootstrap" {}
         "config_update" {}
   "LaunchConfig1" {}
   "MarkLogicVolume1" {}
   "ASG2" {}
   "LaunchConfig2" {}
   "MarkLogicVolume2" {}
   "ElasticLoadBalancer" {}
   "InstanceSecurityGroup" {}
   "InstanceSecurityGroupIngress" {}
 "Outputs" {}
```

```
        "MarkLogic::MetaDataVersion": "2015-07-17-14:49:23",
        "AWS::CloudFormation::Init": {
          "configSets": {
            "default": [
              {"ConfigSet": "bootstrap"},
              {"ConfigSet": "update"}
            ],
            "bootstrap": ["config_bootstrap"],
            "update": ["config_update"]
          },
          "config_bootstrap": {
            "packages": {"yum": {
              "aws-cli": [],
              "rsyslog": [],
              "yum-utils": []
            }},
            "commands": {},
            "services": {"sysvinit": {"MarkLogic": {
              "enabled": "true",
              "ensureRunning": "false"
            }}}
          },
          "config_update": {
            "files": {"/etc/marklogic.conf": {
              "content": {"Fn::Join": [
                "",
                [
                  "MARKLOGIC_CLUSTER_NAME=",
```

/mnt/.../Balisage2015/Presentation/examples/Snippets/multinode-2.j... Format and Indent successful U+007D 520 : 11 Modified 10 new messa...

```
 "Resources" {}
   "MarkLogicDDBTable" {}
   "ASG0" {}
   "LaunchConfig0" {}
   "MarkLogicVolume0" {}
   "ASG1" {}
     "Type" "AWS::AutoScaling::AutoScalingGroup
     "Properties" {}
     "Metadata" {}
       MarkLogic::MetaDataVersion" "2015-07-17-14:49:23
       AWS::CloudFormation::Init" {}
         "configSets" {}
         "config_bootstrap" {}
       config_update" {}
 "LaunchConfig1" {}
 "MarkLogicVolume1" {}
 "ASG2" {}
 "LaunchConfig2" {}
 "MarkLogicVolume2" {}
 "ElasticLoadBalancer" {}
 "InstanceSecurityGroup" {}
 "InstanceSecurityGroupIngress" {}
 "Outputs" {}
```

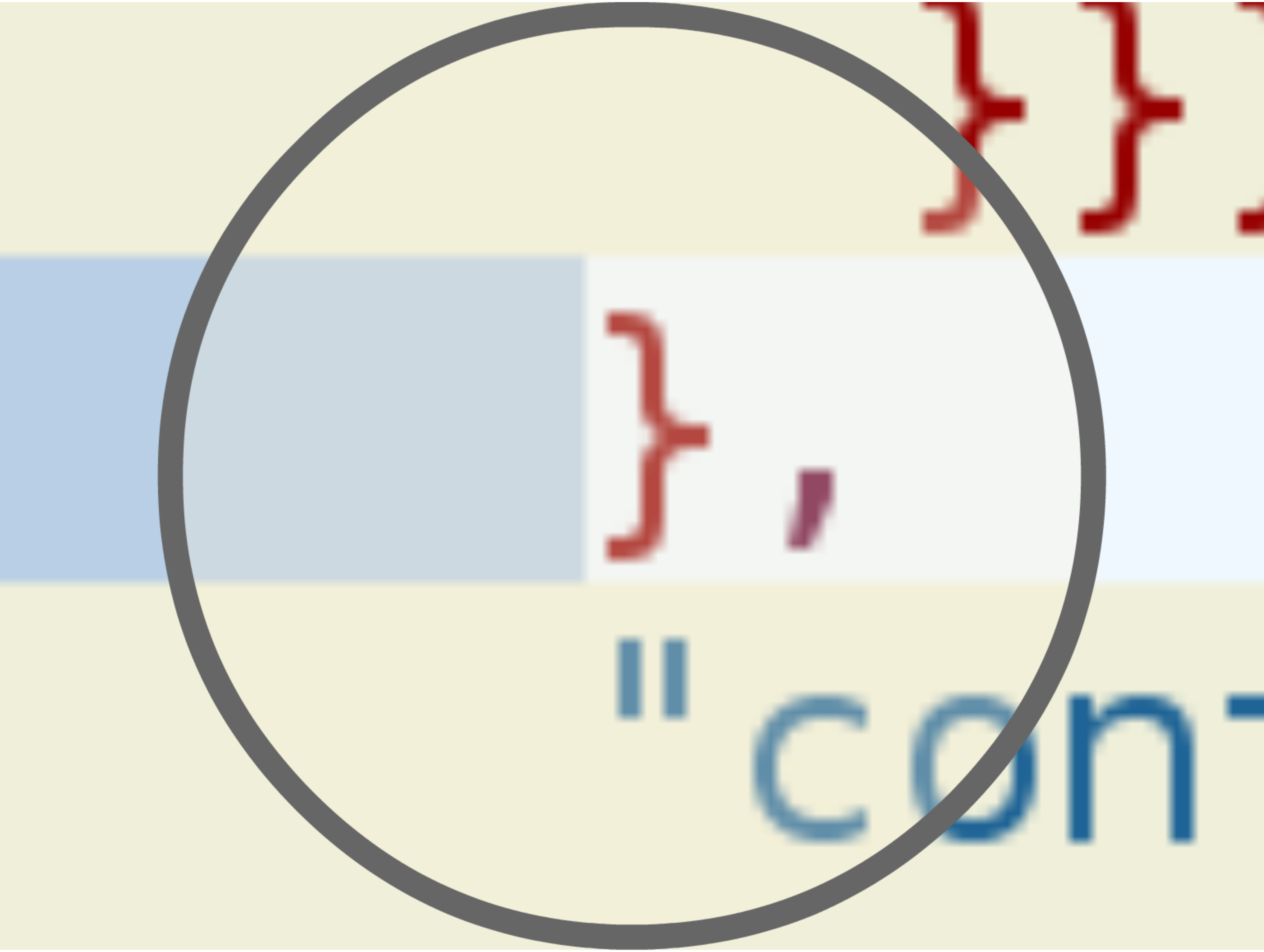
```
              {"ConfigSet": "update"}
            ],
            "bootstrap": ["config_bootstrap"],
            "update": ["config_update"]
          },
          "config_bootstrap": {
            "packages": {"yum": {
              "aws-cli": [],
              "rsyslog": [],
              "yum-utils": []
            }},
            "commands": {},
            "services": {"sysvinit": {"MarkLogic
              "enabled": "true",
              "ensureRunning": "false"
            }}}
          }},
          "config_update": {
            "files": {"/etc/marklogic.conf": {
              "content": {"Fn::Join": [
                "",
                [
                  "MARKLOGIC_CLUSTER_NAME=",
```

/mnt/.../Balisage2015/Presentation/examples/Snippets/multinode-2.j... Format and Indent successful U+0020 521 : 1 Modified 10 new messa...

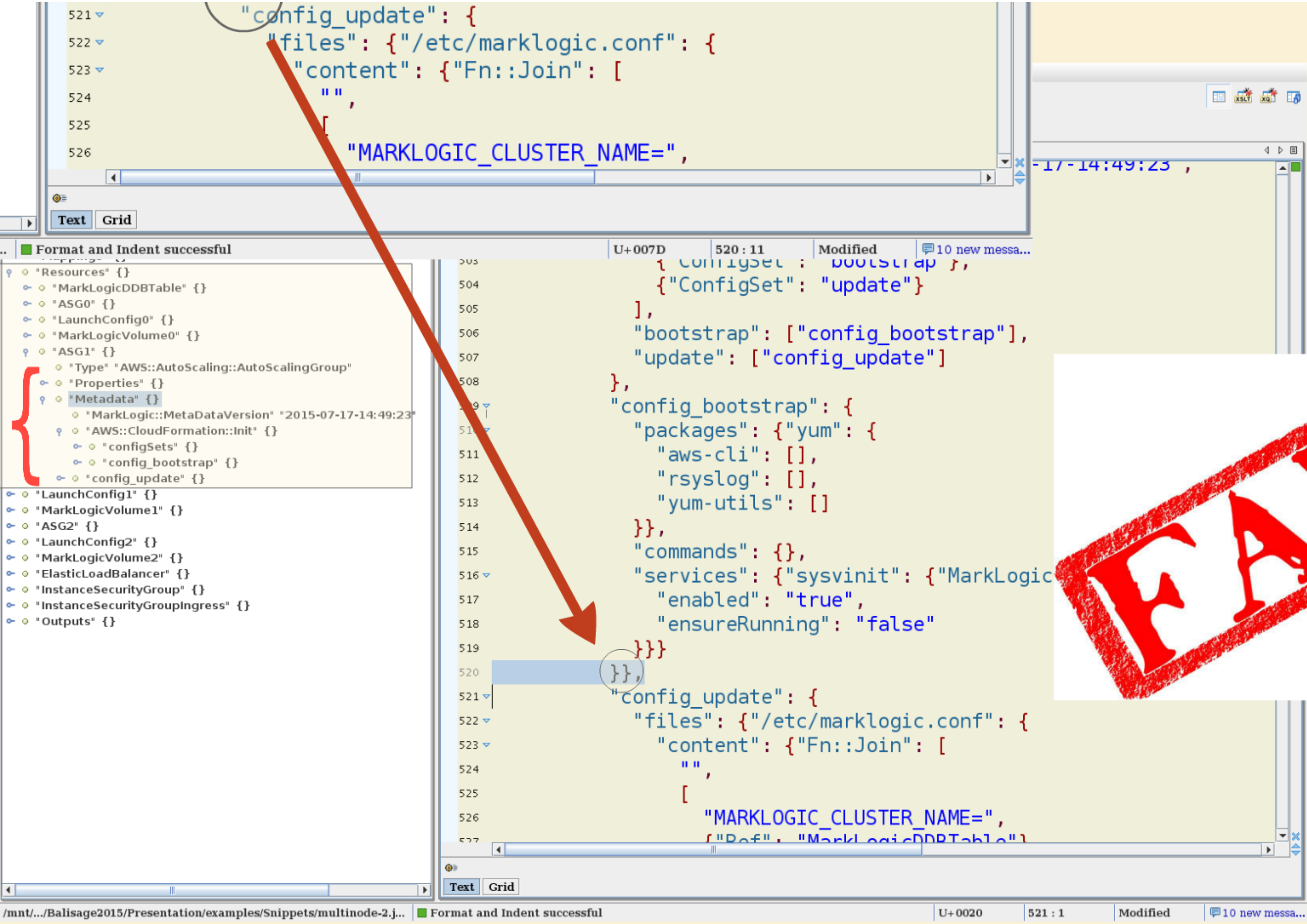
FAIL

#Assumption

```
AutoScalingGroup"

sion" "2015-07-17-14:49:23"
it" {}
```



```
}}
},
"con
```



```
"config_update": {
  "files": {"/etc/marklogic.conf": {
    "content": {"Fn::Join": [
      "",
      [
        "MARKLOGIC_CLUSTER_NAME=",
```

Format and Indent successful | U+007D | 520 : 11 | Modified | 10 new messa...

```
"Resources" {}
  "MarkLogicDDBTable" {}
  "ASG0" {}
  "LaunchConfig0" {}
  "MarkLogicVolume0" {}
  "ASG1" {}
    "Type" "AWS::AutoScaling::AutoScalingGroup"
    "Properties" {}
    "Metadata" {}
      "MarkLogic::MetaDataVersion" "2015-07-17-14:49:23"
      "AWS::CloudFormation::Init" {}
        "configSets" {}
        "config_bootstrap" {}
      "config_update" {}
  "LaunchConfig1" {}
  "MarkLogicVolume1" {}
  "ASG2" {}
  "LaunchConfig2" {}
  "MarkLogicVolume2" {}
  "ElasticLoadBalancer" {}
  "InstanceSecurityGroup" {}
  "InstanceSecurityGroupIngress" {}
  "Outputs" {}
```

```
      { "ConfigSet" : "bootstrap" },
      {"ConfigSet": "update"}
    ],
    "bootstrap": ["config_bootstrap"],
    "update": ["config_update"]
  },
  "config_bootstrap": {
    "packages": {"yum": {
      "aws-cli": [],
      "rsyslog": [],
      "yum-utils": []
    }},
    "commands": {},
    "services": {"sysvinit": {"MarkLogic
      "enabled": "true",
      "ensureRunning": "false"
    }}}
  }},
  "config_update": {
    "files": {"/etc/marklogic.conf": {
      "content": {"Fn::Join": [
        "",
        [
          "MARKLOGIC_CLUSTER_NAME=",
          {"Ref": "MarkLogicDDBTable"},
```

/mnt/.../Balisage2015/Presentation/examples/Snippets/multinode-2.j... | Format and Indent successful | U+0020 | 521 : 1 | Modified | 10 new messa...

JSON

Data - Like

Not Markup

Or Markup Like

http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf

JSON is a lightweight, text-based, language-independent data interchange format.

"Conforming JSON text is a sequence of Unicode code points that strictly conforms to the JSON grammar."

Not Document

#Assumption

FAIL

Not even "doc-ish"

Wrong

A Different Game

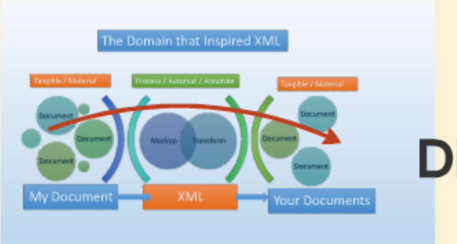
Both

Neither

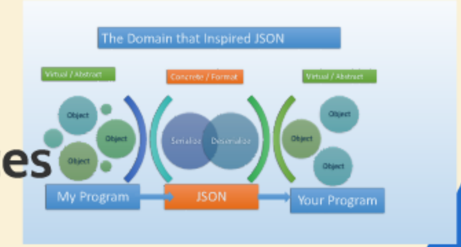
XML

Right

JSON

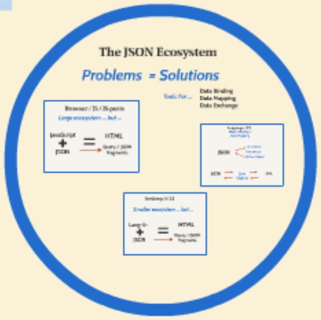


Different Pieces



Preservation of the Representation
"Markup" adds while preserves

Preservation of the Data only
Preservation of Original
Representation an ANTI-GOAL



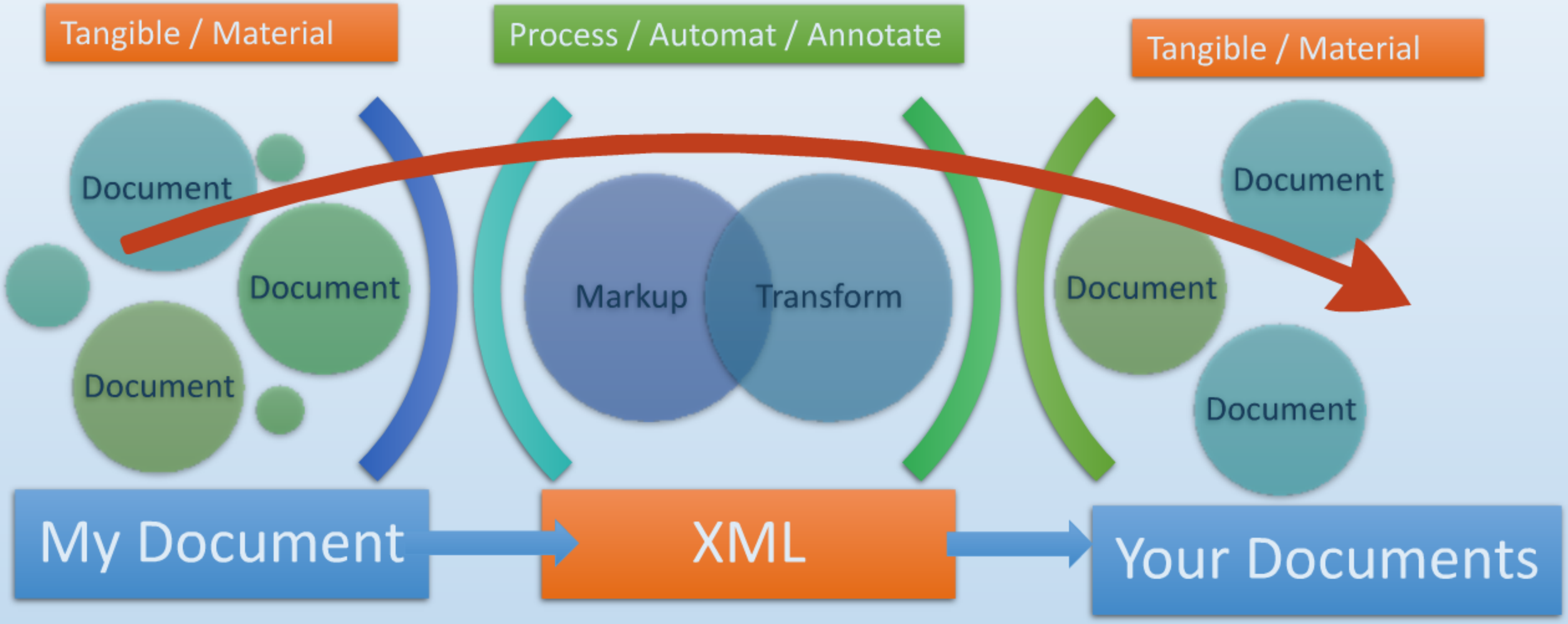
The JSON Ecosystem

Problems = Solutions

Tools For ...

Data Binding
Data Mapping
Data Exchange

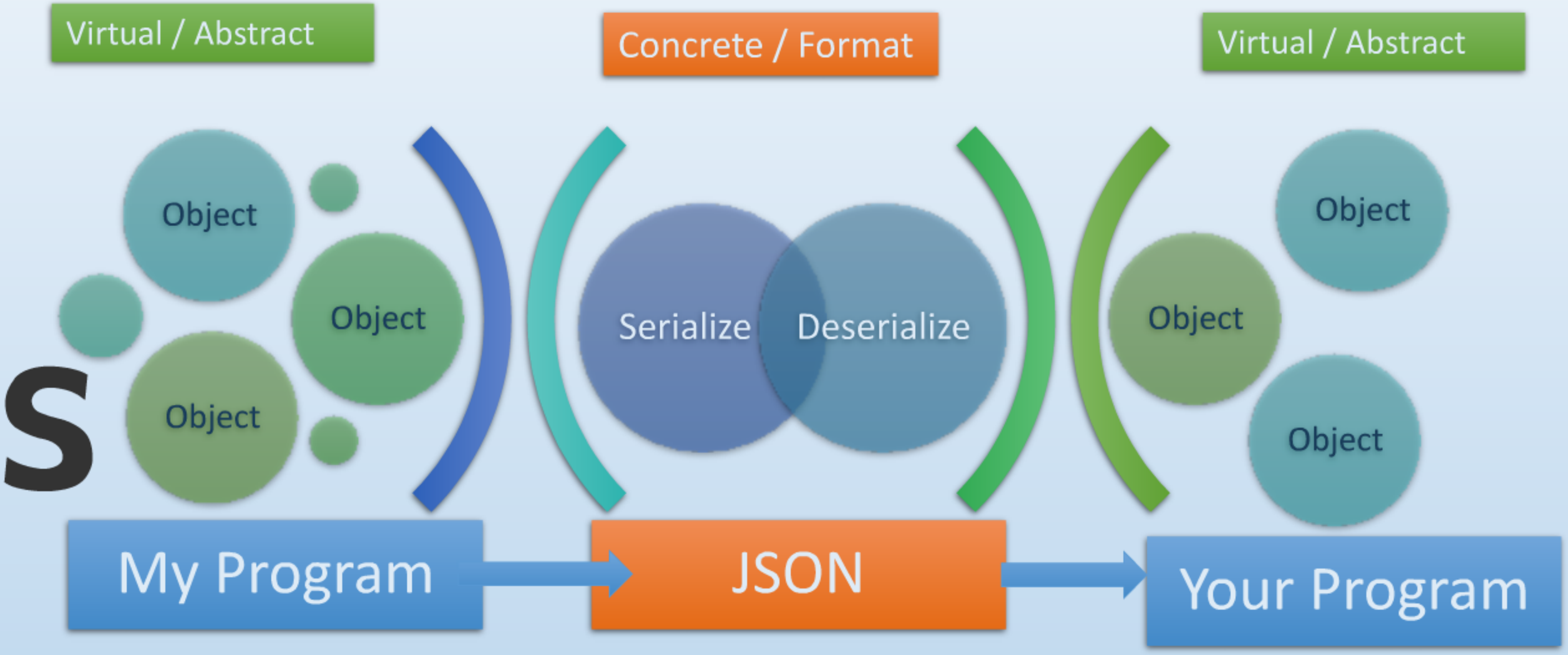
The Domain that Inspired XML



Preservation of the Representation

es

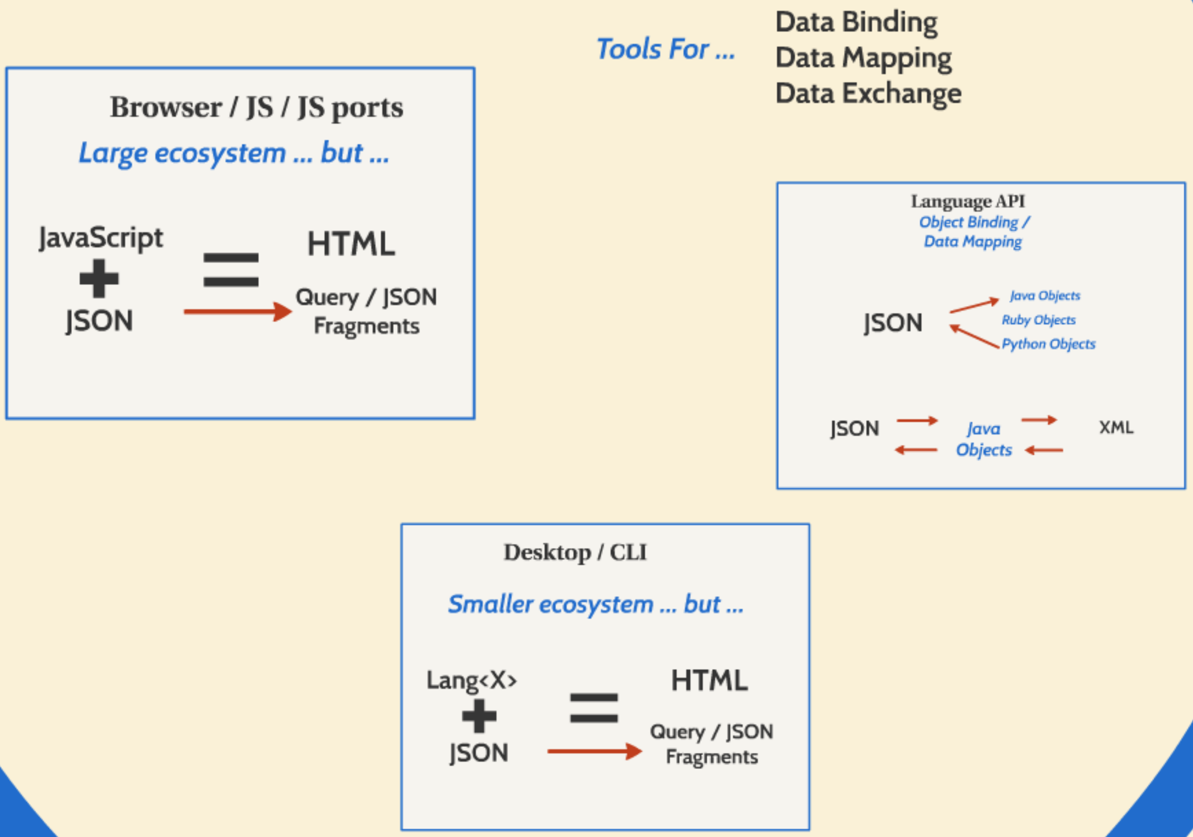
The Domain that Inspired JSON



Preservation of the Data only

The JSON Ecosystem

Problems = Solutions



Tools For ... Data Binding
Data Mapping
Data Exchange

Browser / JS / JS ports

Large ecosystem ... but ...

JavaScript + JSON = HTML → Query / JSON Fragments

Language API

Object Binding / Data Mapping

JSON → *Java Objects*
JSON ← *Ruby Objects*
JSON ← *Python Objects*

JSON → *Java Objects* → XML
JSON ← *Java Objects* ← XML

Desktop / CLI

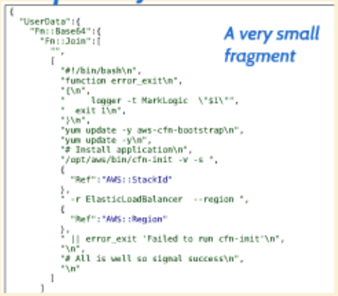
Smaller ecosystem ... but ...

Lang<X> + JSON = HTML → Query / JSON Fragments

Concept:

Continuous Refactoring

Required JSON Format



```
{
  "UserData":{
    "Fn::Base64":{
      "Fn::Join":[
        "",
        [
          "#!/bin/bash\n",
          "function error_exit\n",
          "{\n",
          "      logger -t MarkLogic  \"$1\"",
          "  exit 1\n",
          "}\n",
          "yum update -y aws-cfn-bootstrap\n",
          "yum update -y\n",
          "# Install application\n",
          "/opt/aws/bin/cfn-init -v -s ",
          {
            "Ref":"AWS::StackId"
          },
          " -r ElasticLoadBalancer  --region ",
          {
            "Ref":"AWS::Region"
          },
          " || error_exit 'Failed to run cfn-init'\n",
          "\n",
          "# All is well so signal success\n",
          "\n"
        ]
      ]
```

A very small fragment

From JSON to XML ?

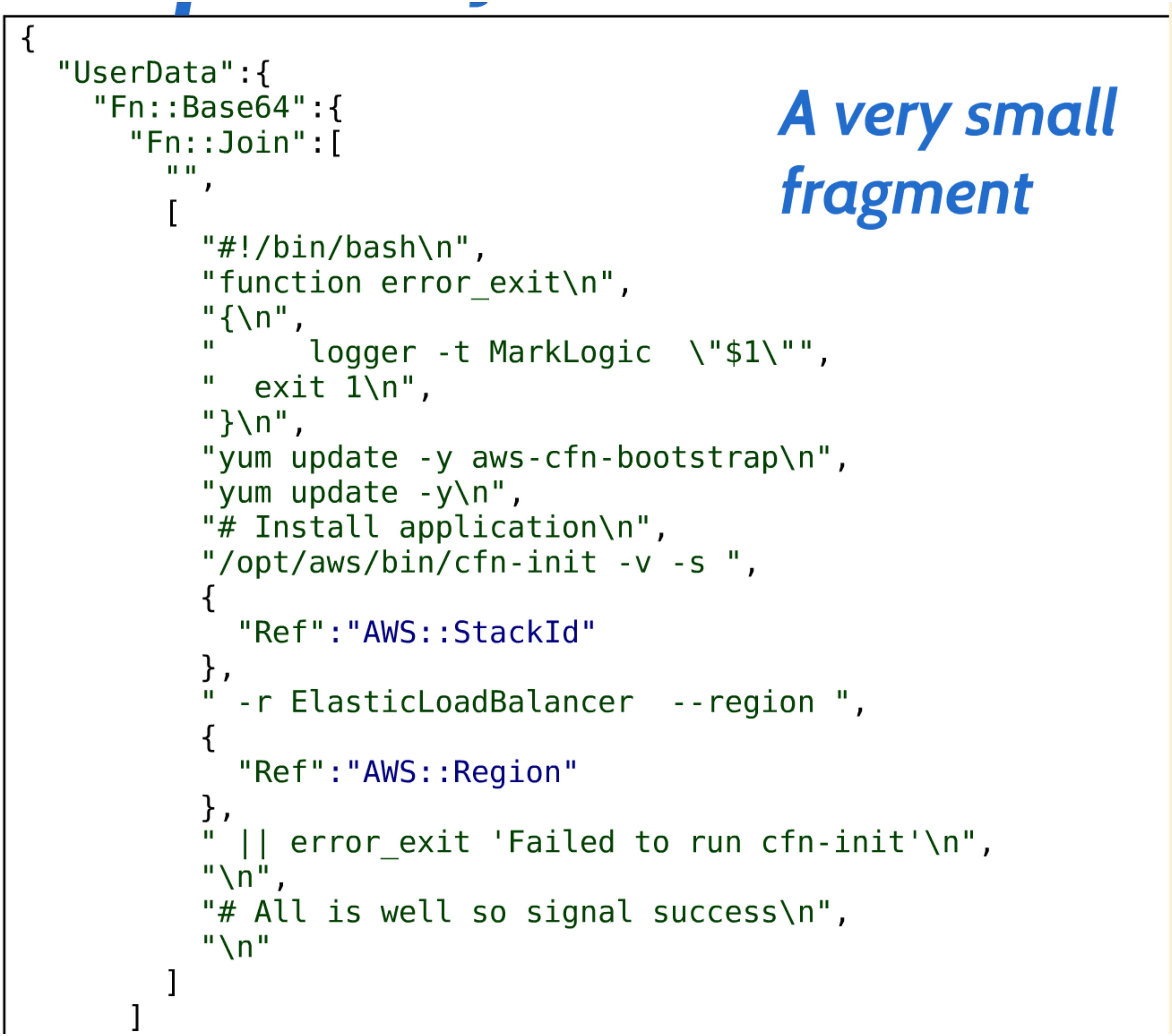
Typical "JSON to XML" conversion

Refactor #2

JXML

Refactor #1

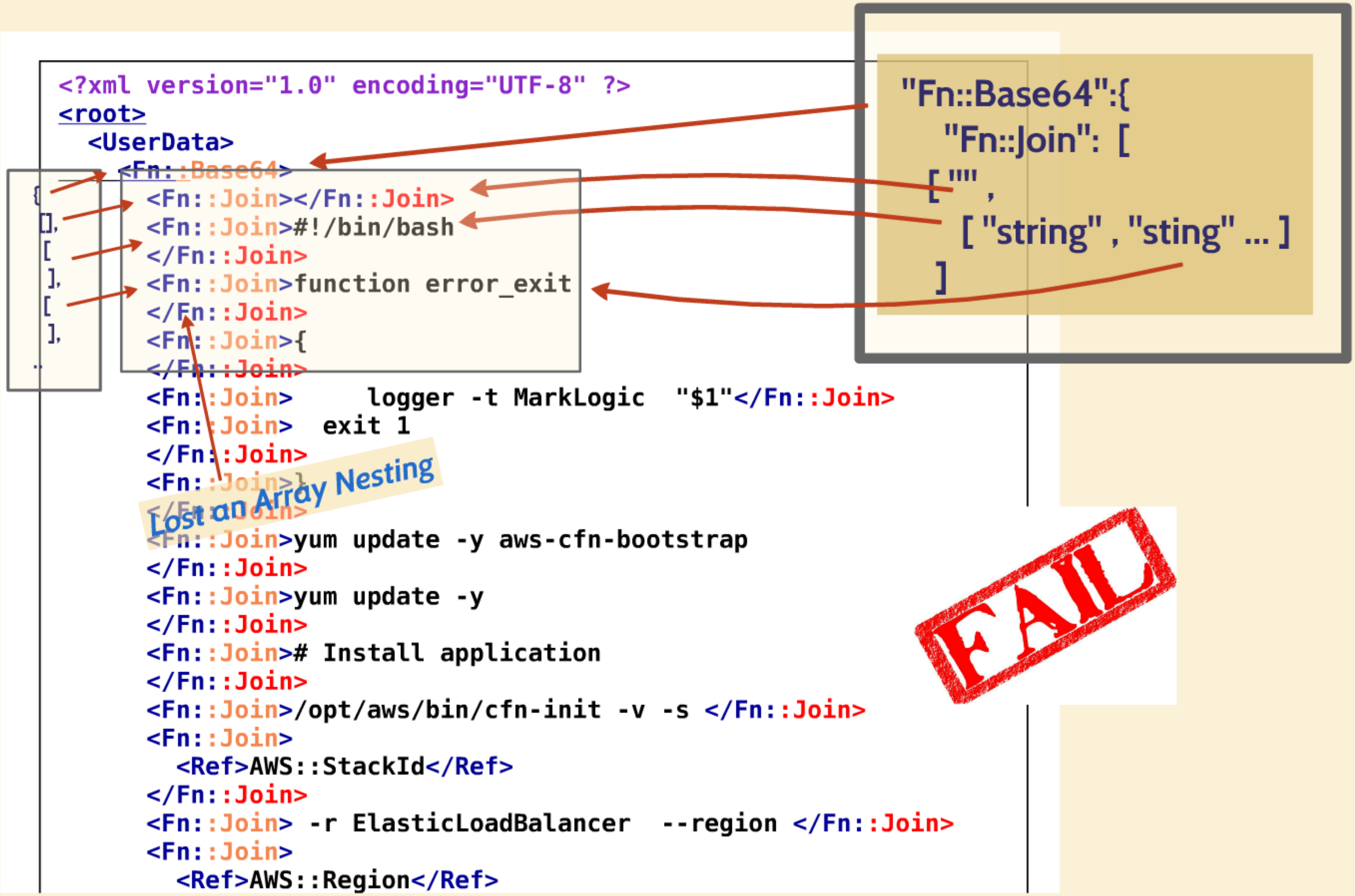
JSONx



```
{
  "UserData":{
    "Fn::Base64":{
      "Fn::Join":[
        "",
        [
          "#!/bin/bash\n",
          "function error_exit\n",
          "{\n",
          "     logger -t MarkLogic  \"$1\"",
          "  exit 1\n",
          "}\n",
          "yum update -y aws-cfn-bootstrap\n",
          "yum update -y\n",
          "# Install application\n",
          "/opt/aws/bin/cfn-init -v -s ",
          {
            "Ref":"AWS::StackId"
          },
          " -r ElasticLoadBalancer  --region ",
          {
            "Ref":"AWS::Region"
          },
          " || error_exit 'Failed to run cfn-init'\n",
          "\n",
          "# All is well so signal success\n",
          "\n"
        ]
      ]
```

A very small fragment

Typical "JSON to XML" conversion



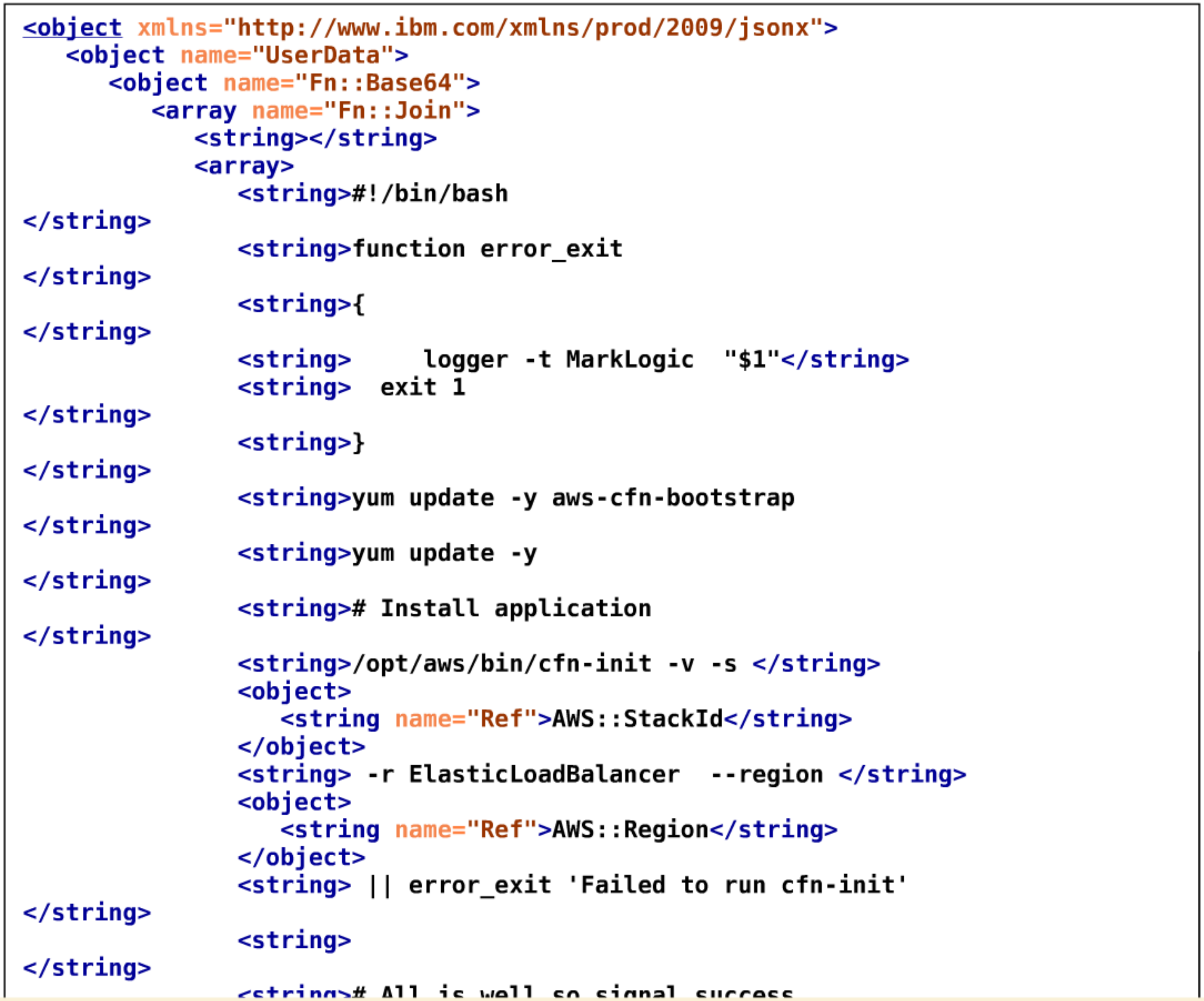
```
<?xml version="1.0" encoding="UTF-8" ?>
<root>
  <UserData>
    <Fn::Base64>
      <Fn::Join></Fn::Join>
      <Fn::Join>#!/bin/bash
      </Fn::Join>
      <Fn::Join>function error_exit
      </Fn::Join>
      <Fn::Join>{
      </Fn::Join>
      <Fn::Join>     logger -t MarkLogic  "$1"</Fn::Join>
      <Fn::Join>  exit 1
      </Fn::Join>
      <Fn::Join>}
      </Fn::Join>
      <Fn::Join>yum update -y aws-cfn-bootstrap
      </Fn::Join>
      <Fn::Join>yum update -y
      </Fn::Join>
      <Fn::Join># Install application
      </Fn::Join>
      <Fn::Join>/opt/aws/bin/cfn-init -v -s </Fn::Join>
      <Fn::Join>
        <Ref>AWS::StackId</Ref>
      </Fn::Join>
      <Fn::Join> -r ElasticLoadBalancer  --region </Fn::Join>
      <Fn::Join>
        <Ref>AWS::Region</Ref>
```

```
{
 [],
 [
  ],
 [
  ],
..
```

```
"Fn::Base64":{
   "Fn::Join": [
  [ "" ,
    [ "string" , "sting" ... ]
  ]
```

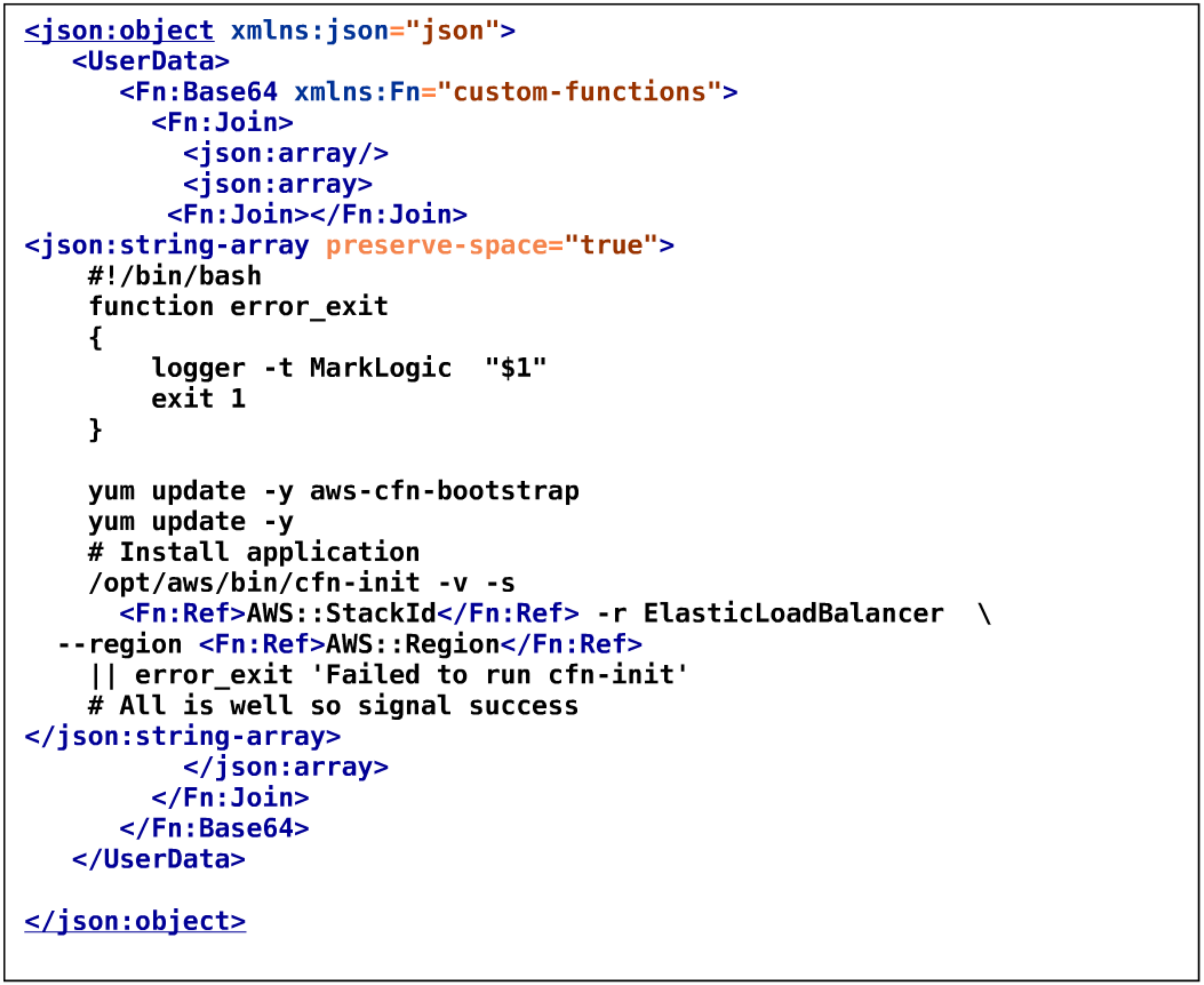
http://codebeautify.org/jsontoxml

JSONx



```
<object xmlns="http://www.ibm.com/xmlns/prod/2009/jsonx">
   <object name="UserData">
      <object name="Fn::Base64">
         <array name="Fn::Join">
            <string></string>
            <array>
               <string>#!/bin/bash
</string>
               <string>function error_exit
</string>
               <string>{
</string>
               <string>     logger -t MarkLogic  "$1"</string>
               <string>  exit 1
</string>
               <string>}
</string>
               <string>yum update -y aws-cfn-bootstrap
</string>
               <string>yum update -y
</string>
               <string># Install application
</string>
               <string>/opt/aws/bin/cfn-init -v -s </string>
               <object>
                  <string name="Ref">AWS::StackId</string>
               </object>
               <string> -r ElasticLoadBalancer  --region </string>
               <object>
                  <string name="Ref">AWS::Region</string>
               </object>
               <string> || error_exit 'Failed to run cfn-init'
</string>
               <string>
</string>
               <string># All is well so signal success
```

Refactor #1



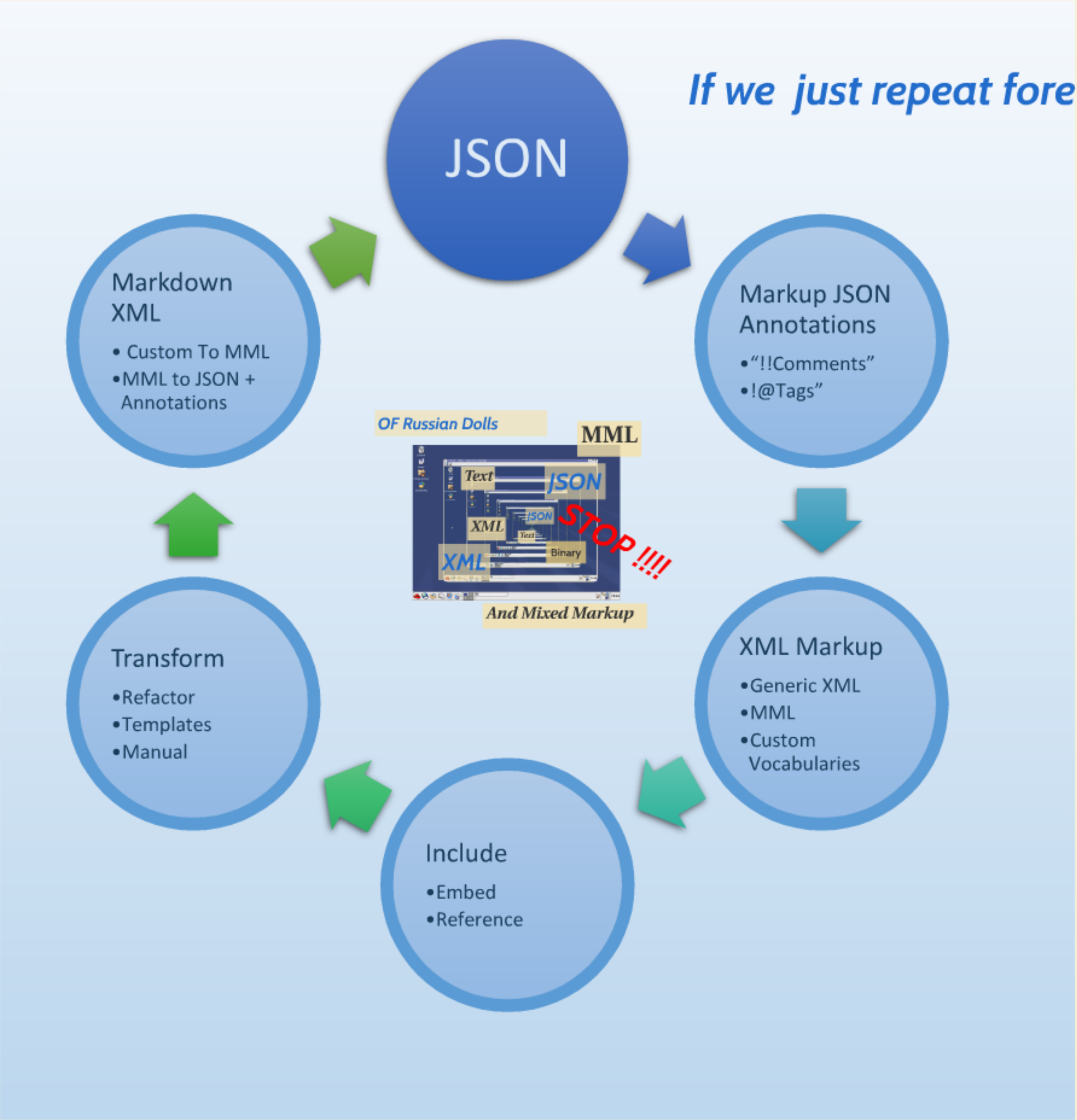
```
<json:object xmlns:json="json">
   <UserData>
      <Fn:Base64 xmlns:Fn="custom-functions">
        <Fn:Join>
          <json:array/>
          <json:array>
         <Fn:Join></Fn:Join>
<json:string-array preserve-space="true">
    #!/bin/bash
    function error_exit
    {
        logger -t MarkLogic  "$1"
        exit 1
    }

    yum update -y aws-cfn-bootstrap
    yum update -y
    # Install application
    /opt/aws/bin/cfn-init -v -s
      <Fn:Ref>AWS::StackId</Fn:Ref> -r ElasticLoadBalancer  \
  --region <Fn:Ref>AWS::Region</Fn:Ref>
    || error_exit 'Failed to run cfn-init'
    # All is well so signal success
</json:string-array>
          </json:array>
        </Fn:Join>
      </Fn:Base64>
   </UserData>

</json:object>
```

Refactor #2

```
<UserData encoding="base64" space="preserve">
    #!/bin/bash
    function error_exit
    {
        logger -t MarkLogic  "$1"
        exit 1
    }

    yum update -y aws-cfn-bootstrap
    yum update -y
    # Install application
    /opt/aws/bin/cfn-init -v -s
      <ref idref="AWS::StackId"/> -r ElasticLoadBalancer  \
  --region <ref idref="AWS::Region"/>
    || error_exit 'Failed to run cfn-init'
    # All is well so signal success
</UserData>
```

A good theory ...

If we just repeat forever ...



JSON

Markup JSON Annotations
- "!!Comments"
- !@Tags"

XML Markup
- Generic XML
- MML
- Custom Vocabularies

Include
- Embed
- Reference

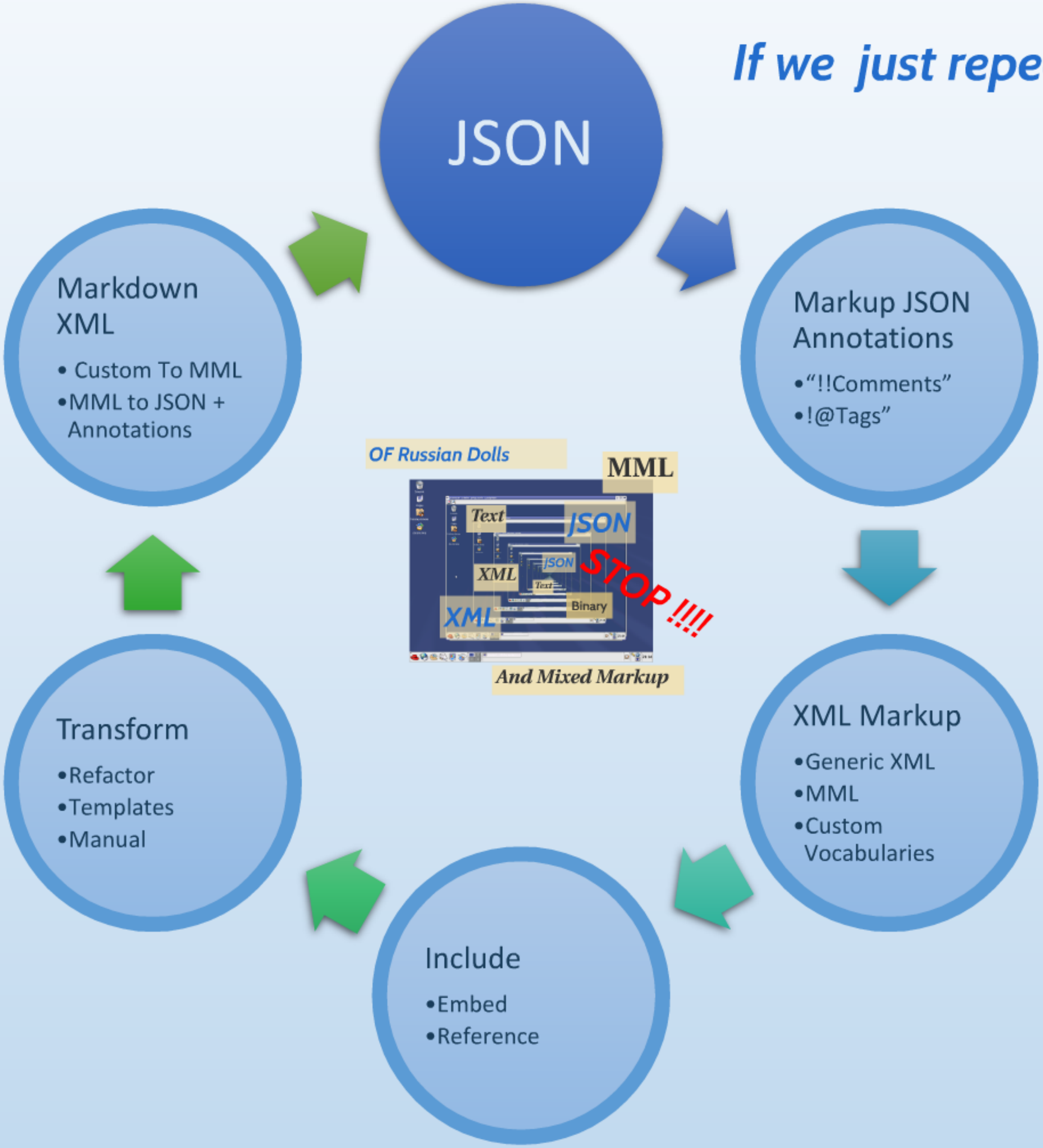
Transform
- Refactor
- Templates
- Manual

Markdown XML
- Custom To MML
- MML to JSON + Annotations

OF Russian Dolls

And Mixed Markup

If we just repeat forever …



JSON

Markup JSON Annotations

- "!!Comments"
- !@Tags"

XML Markup

- Generic XML
- MML
- Custom Vocabularies

Include

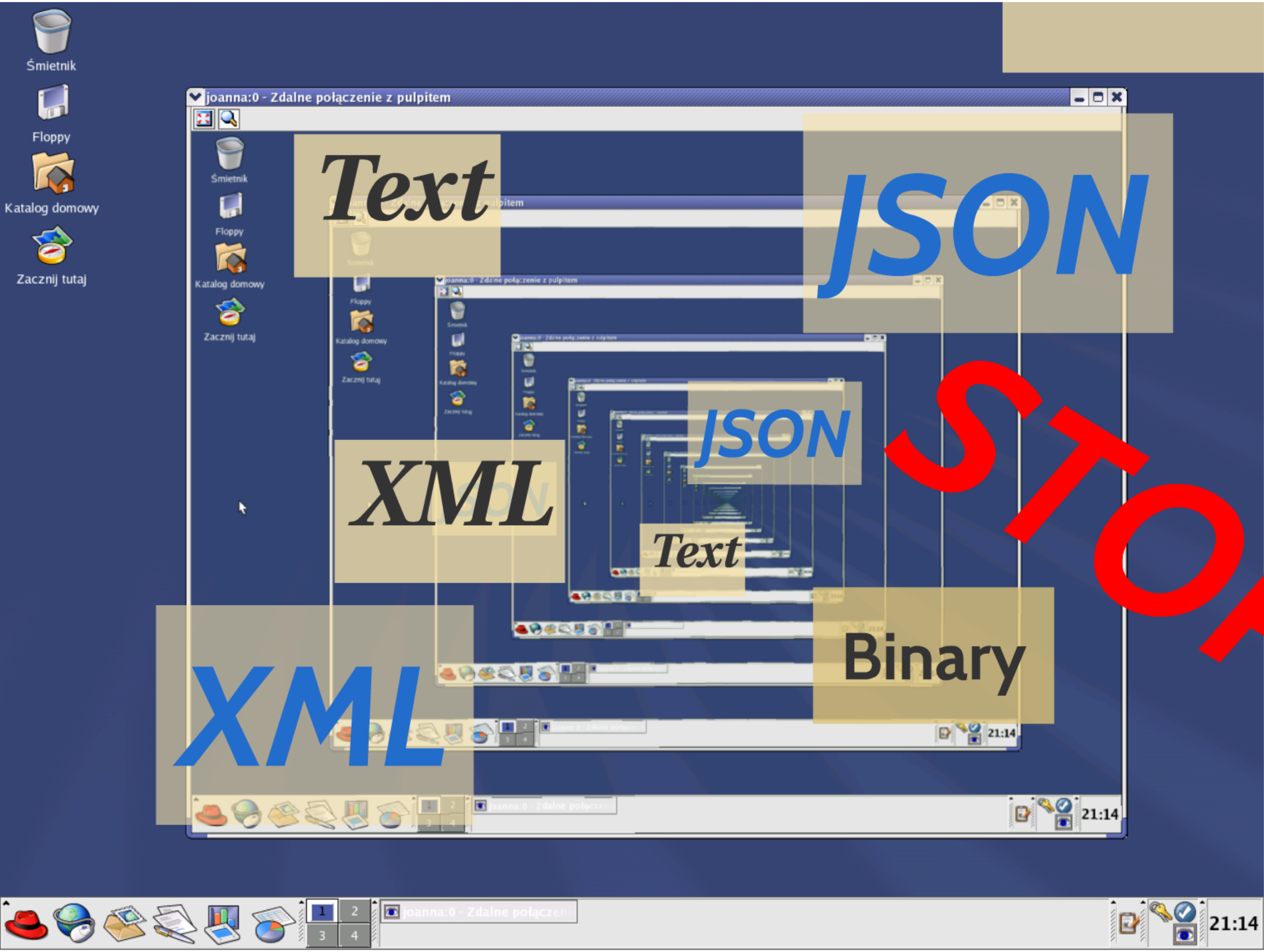
- Embed
- Reference

Transform

- Refactor
- Templates
- Manual

Markdown XML

- Custom To MML
- MML to JSON + Annotations



Text

JSON

XML

JSON

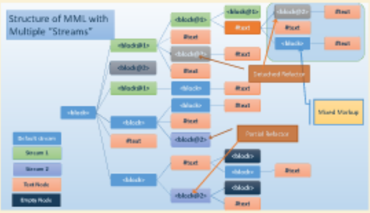
Text

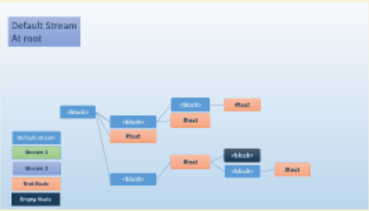
STOP

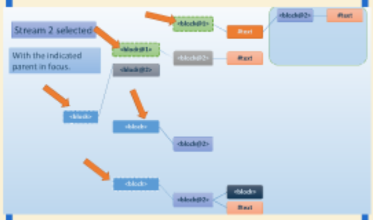
XML

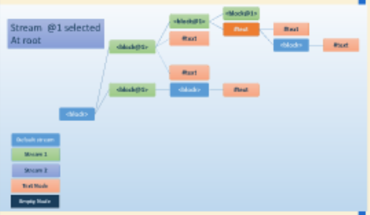
Binary

Concept: "Streams"

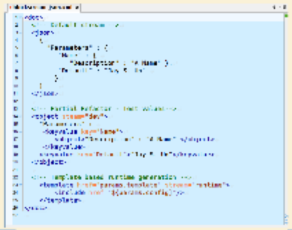






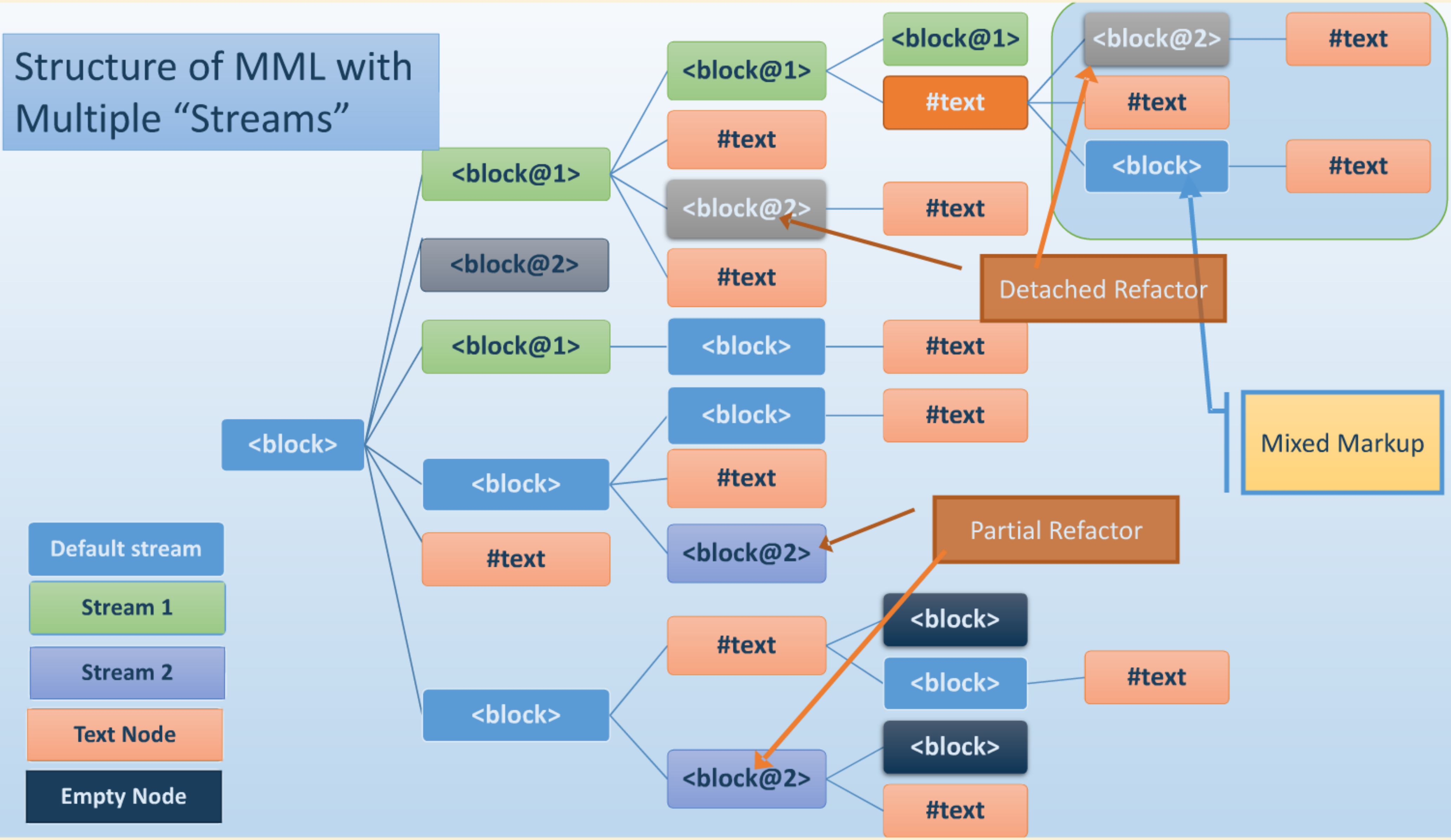


Example



```
<doc>
  <!-- Default stream -->
  <json>
    {
      "Parameters" : {
        "Name" : {
          "Description" : "A Name" },
        [illegible]
      }
    }
  </json>

  <!-- Partial Refactor - Text value-->
  <object stream="dev">
    [illegible]
     <keyvalue key="Name">
        [illegible]
     </keyvalue>
    [illegible]
  </object>

  <!-- Template based runtime generation -->
    <template href="params.template" stream="runtime">
        <include href="${params.config}"/>
    </template>
</doc>
```

Structure of MML with Multiple "Streams"



<block>

<block@1>

<block@1>

<block@1>

#text

<block@2>

#text

#text

<block@2>

#text

#text

<block>

#text

<block@2>

<block@1>

<block>

#text

<block>

<block>

#text

#text

<block@2>

#text

<block>

#text

<block>

<block>

#text

<block@2>

<block>

#text

Detached Refactor

Mixed Markup

Partial Refactor

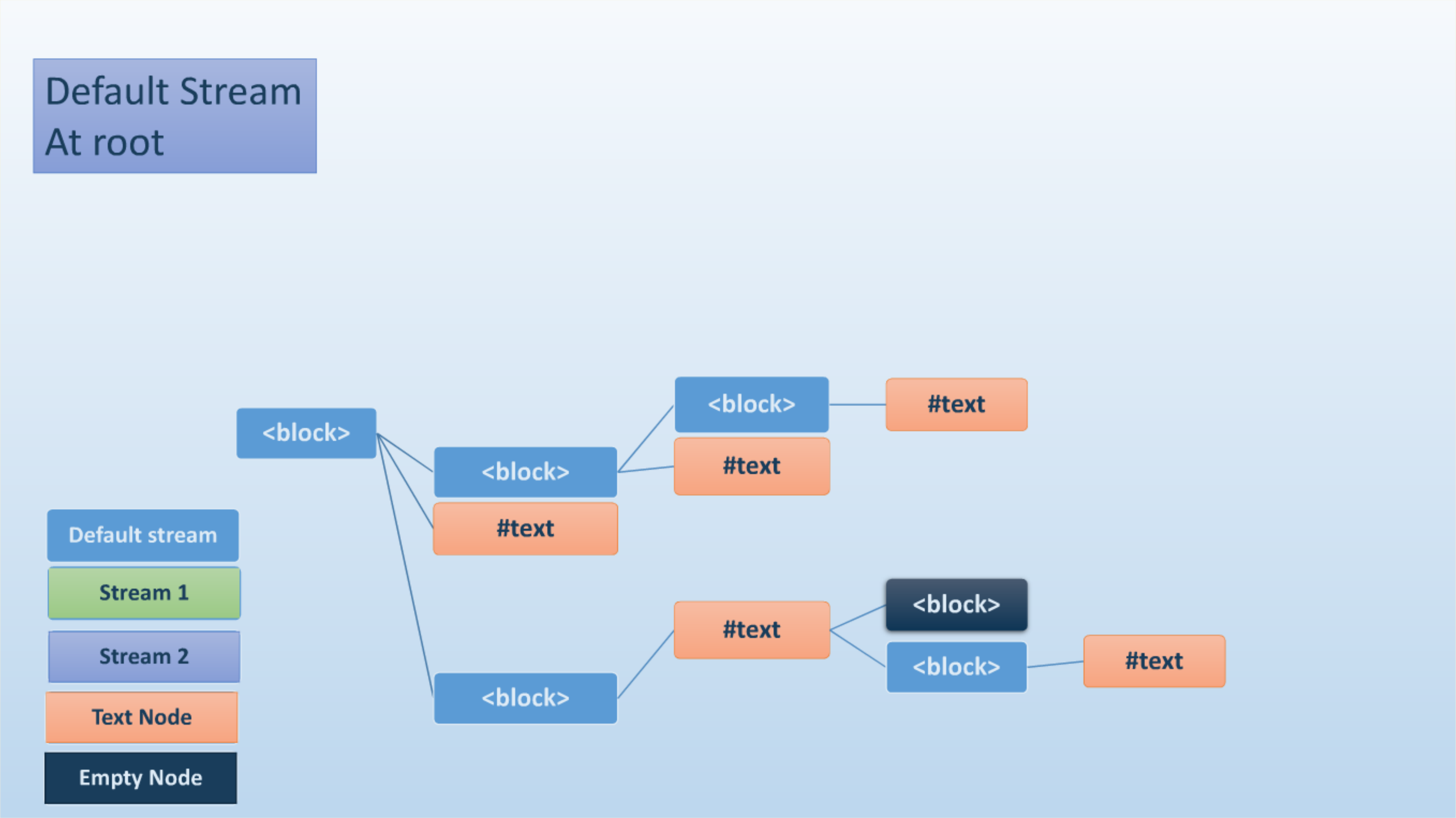
Default stream

Stream 1

Stream 2

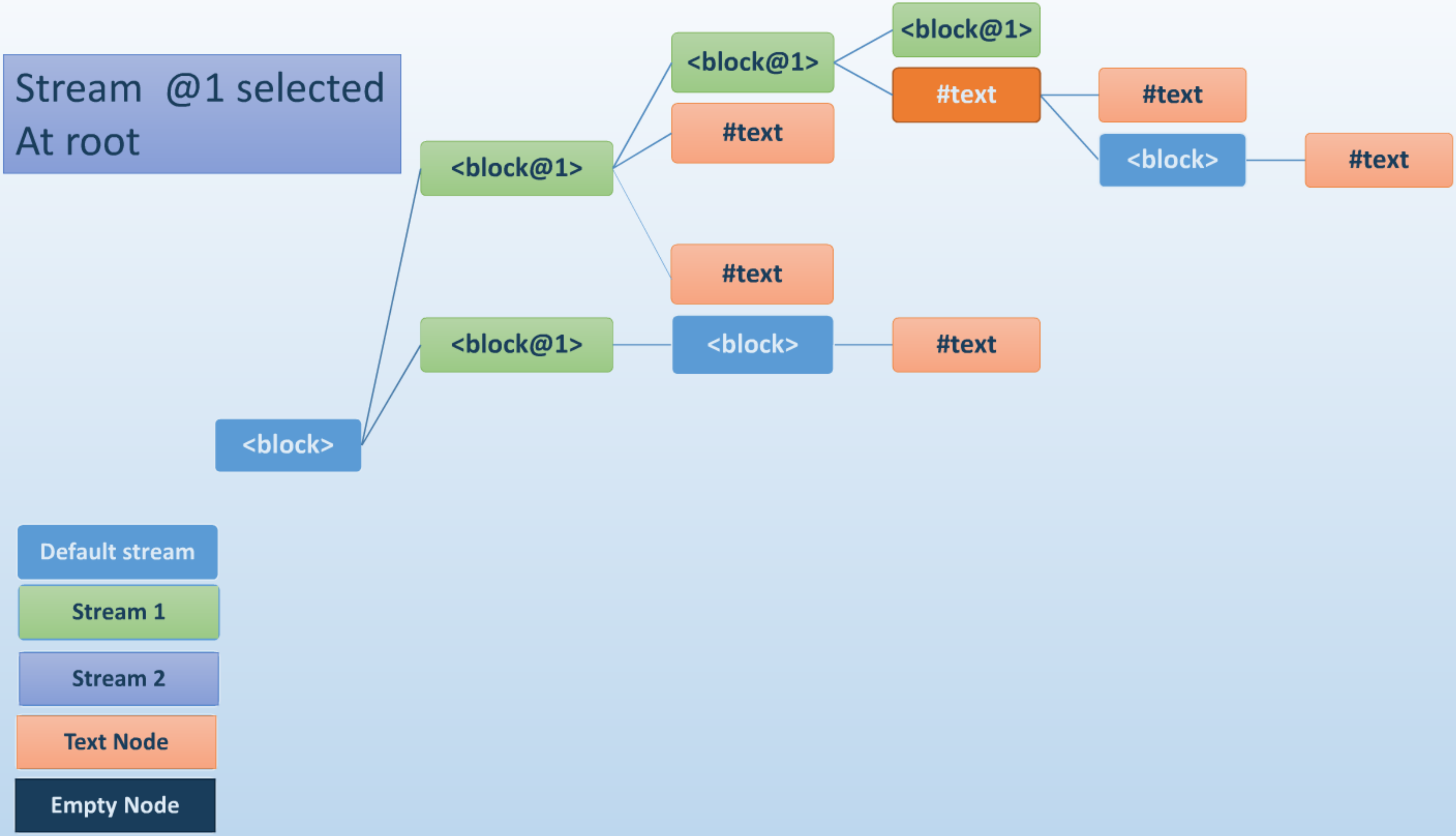
Text Node

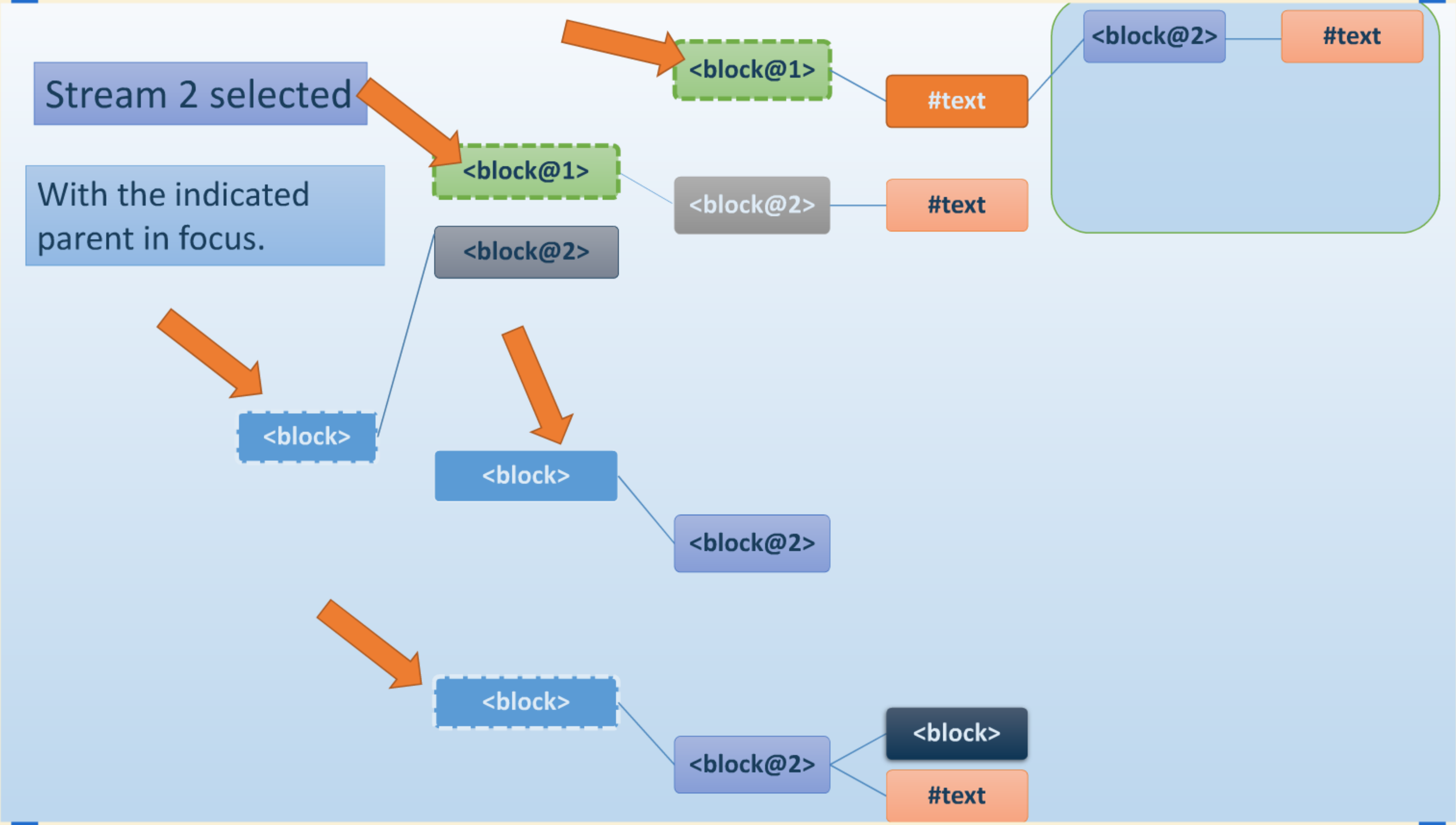
Empty Node



Default Stream
At root

Stream @1 selected
At root

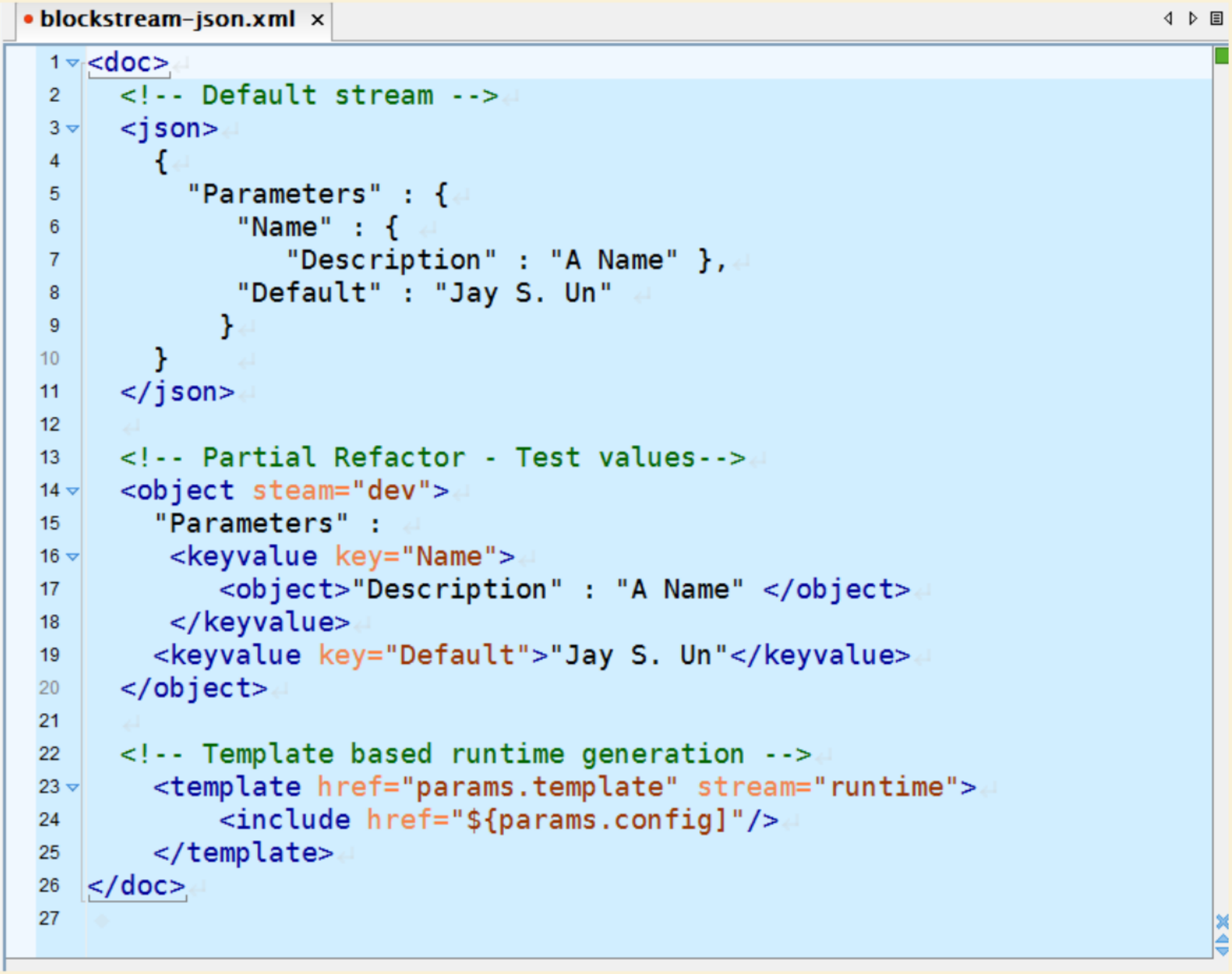




Stream 2 selected

With the indicated parent in focus.

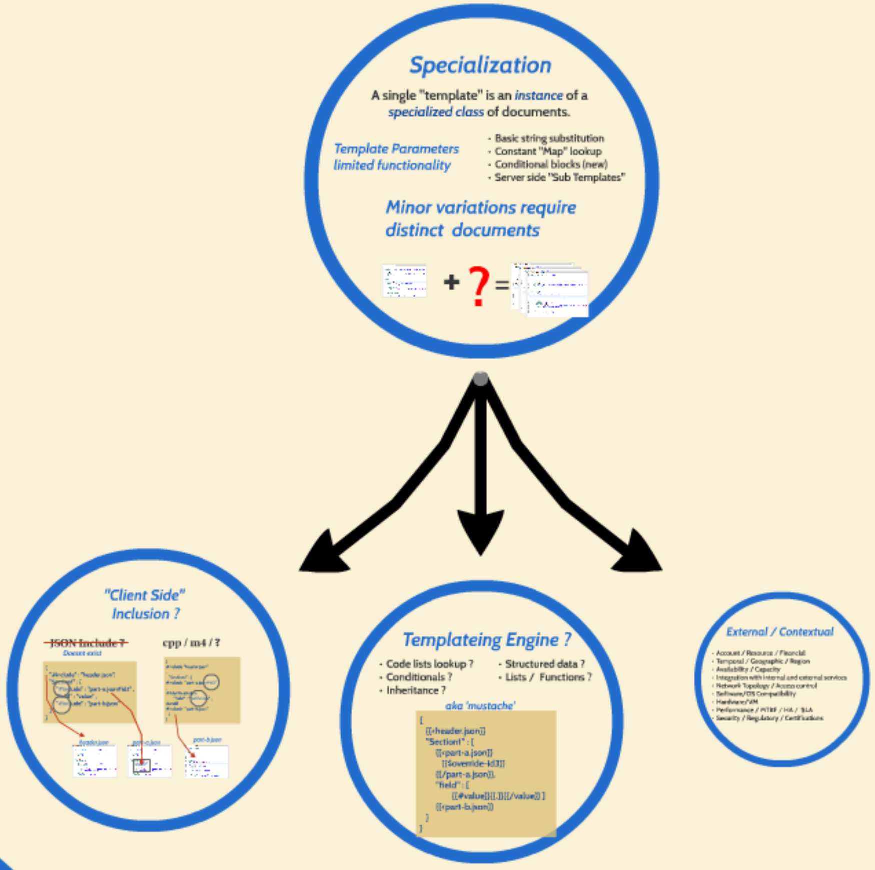
Example



```
<doc>
  <!-- Default stream -->
  <json>
    {
      "Parameters" : {
         "Name" : {
            "Description" : "A Name" },
         "Default" : "Jay S. Un"
        }
    }
  </json>

  <!-- Partial Refactor - Test values-->
  <object steam="dev">
    "Parameters" :
     <keyvalue key="Name">
        <object>"Description" : "A Name" </object>
     </keyvalue>
    <keyvalue key="Default">"Jay S. Un"</keyvalue>
  </object>

  <!-- Template based runtime generation -->
    <template href="params.template" stream="runtime">
        <include href="${params.config}"/>
    </template>
</doc>
```

Concept:

"Meta Templates"



Specialization

A single "template" is an *instance* of a *specialized class* of documents.

Template Parameters limited functionality

- Basic string substitution
- Constant "Map" lookup
- Conditional blocks (new)
- Server side "Sub Templates"

Minor variations require distinct documents

+ ? =

"Client Side" Inclusion ?

~~JSON Include ?~~
Doesnt exist
"#include" : "header.json"
"#include" : "part-a.json#id3"
"value"
"#include" : "part-b.json"
cpp / m4 / ?
header.json
part-a.json
part-b.json

Templateing Engine ?

- Code lists lookup ?
- Conditionals ?
- Inheritance ?
- Structured data ?
- Lists / Functions ?

aka 'mustache'

```
{
  {{<header.json}}
  "Section1" : [
     {{<part-a.json}}
       {{$override-id3}}
     {{/part-a.json}},
     "field" : [
           {{#value}}{{.}}{{/value}} ]
     {{<part-b.json}}
  ]
}
```

External / Contextual

- Account / Resource / Financial
- Temporal / Geographic / Region
- Availability / Capacity
- Integration with internal and external services
- Network Topology / Access control
- Software/OS Compatibility
- Hardware/VM
- Performance / MTBF / HA / SLA
- Security / Regulatory / Certifications

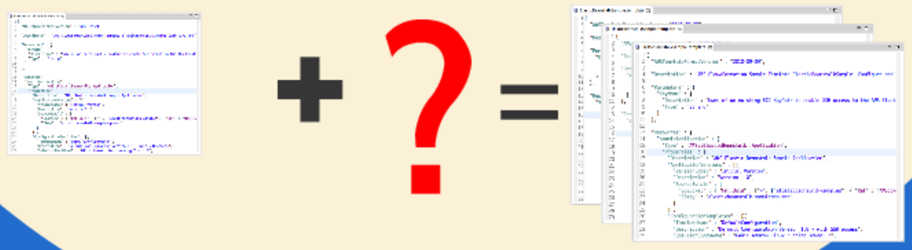
Specialization

A single "template" is an *instance* of a *specialized class* of documents.

Template Parameters limited functionality

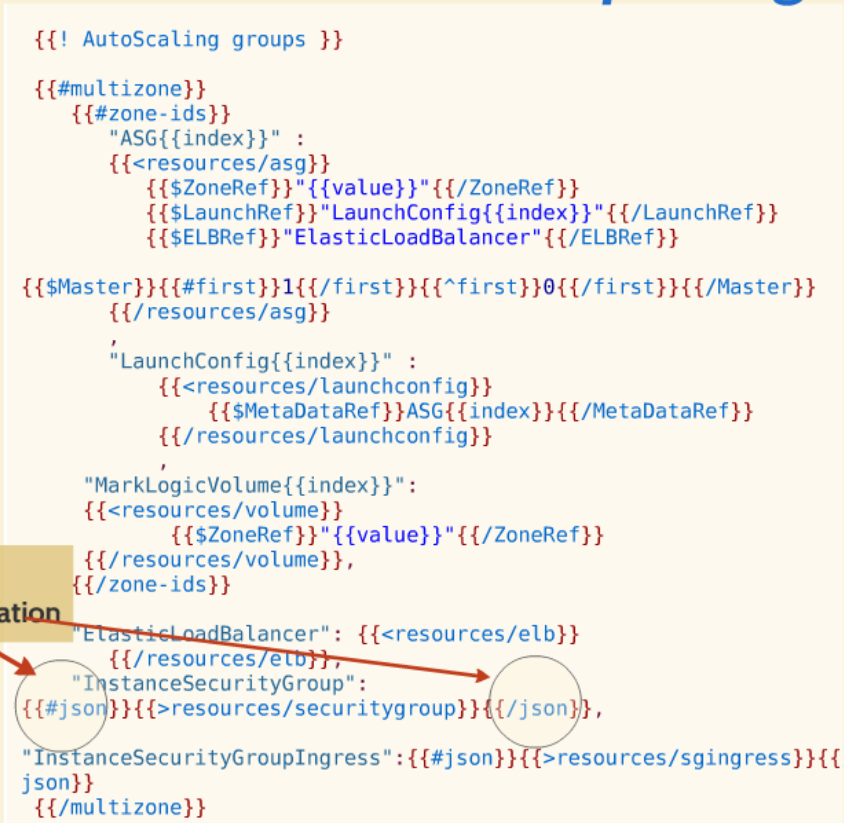
- Basic string substitution
- Constant "Map" lookup
- Conditional blocks (new)
- Server side "Sub Templates"

Minor variations require distinct documents



A working implementation

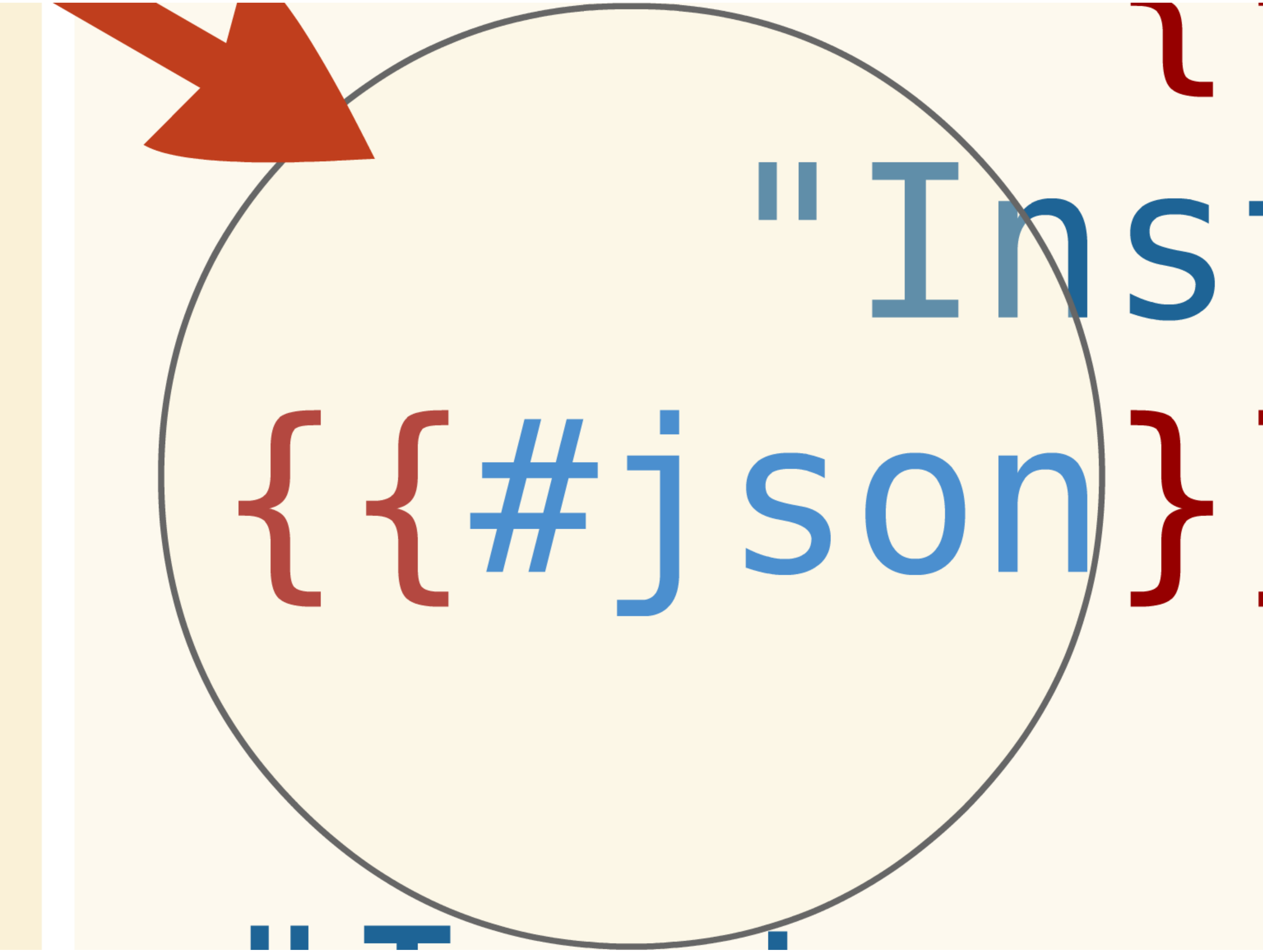
Mustached based templating



```
 {{! AutoScaling groups }}

 {{#multizone}}
    {{#zone-ids}}
       "ASG{{index}}" :
       {{<resources/asg}}
          {{$ZoneRef}}"{{value}}"{{/ZoneRef}}
          {{$LaunchRef}}"LaunchConfig{{index}}"{{/LaunchRef}}
          {{$ELBRef}}"ElasticLoadBalancer"{{/ELBRef}}

{{$Master}}{{#first}}1{{/first}}{{^first}}0{{/first}}{{/Master}}
       {{/resources/asg}}
       ,
       "LaunchConfig{{index}}" :
           {{<resources/launchconfig}}
               {{$MetaDataRef}}ASG{{index}}{{/MetaDataRef}}
           {{/resources/launchconfig}}
           ,
     "MarkLogicVolume{{index}}":
     {{<resources/volume}}
            {{$ZoneRef}}"{{value}}"{{/ZoneRef}}
     {{/resources/volume}},
    {{/zone-ids}}

    "ElasticLoadBalancer": {{<resources/elb}}
       {{/resources/elb}},
    "InstanceSecurityGroup":
{{#json}}{{>resources/securitygroup}}{{/json}},

"InstanceSecurityGroupIngress":{{#json}}{{>resources/sgingress}}{{
json}}
 {{/multizone}}
```

Critical !
Inline Validation

Improvement?



```
                 {
          "Ins
{{#json}}
"I
```

POC Implementation

- XML Wrapped Content
- Open and Extensible base schema
- "Agile" Derived Schemas at will

Oxygen Plugin / Extension
Interactive refactoring
Multi Streamed

POC Implementation

- XML Wrapped Content
- Open and Extensible base schema
- "Agile" Derived Schemas at will

Oxygen Plugin / Extension
Interactive refactoring
Multi Streamed

Agile - Derived Schemas at Will

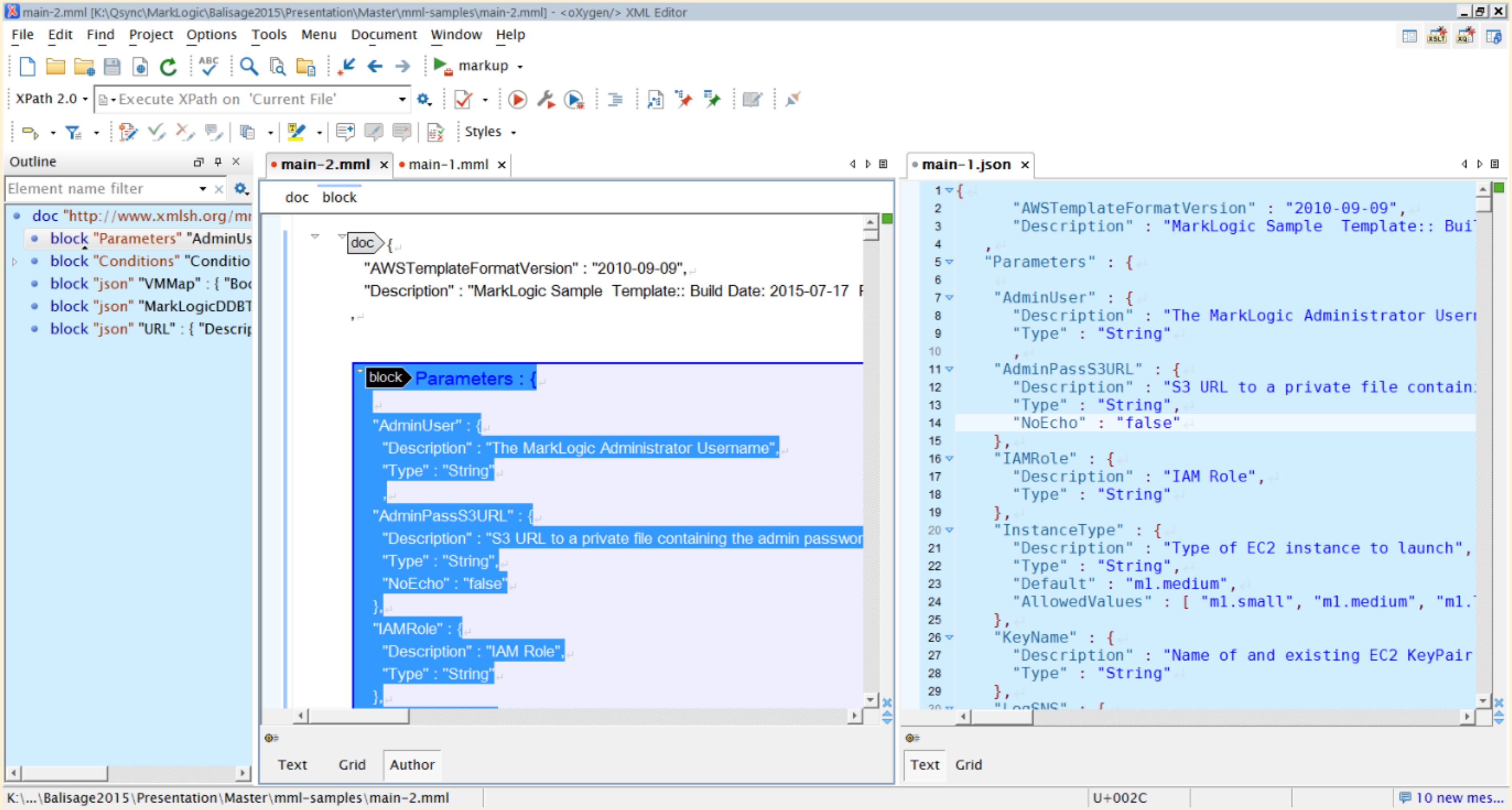
Oxygen Plugin / Extension

Interactive refactoring

Multi Streamed

"Late Breaking" -- About 45 minutes ago 1000 miles away

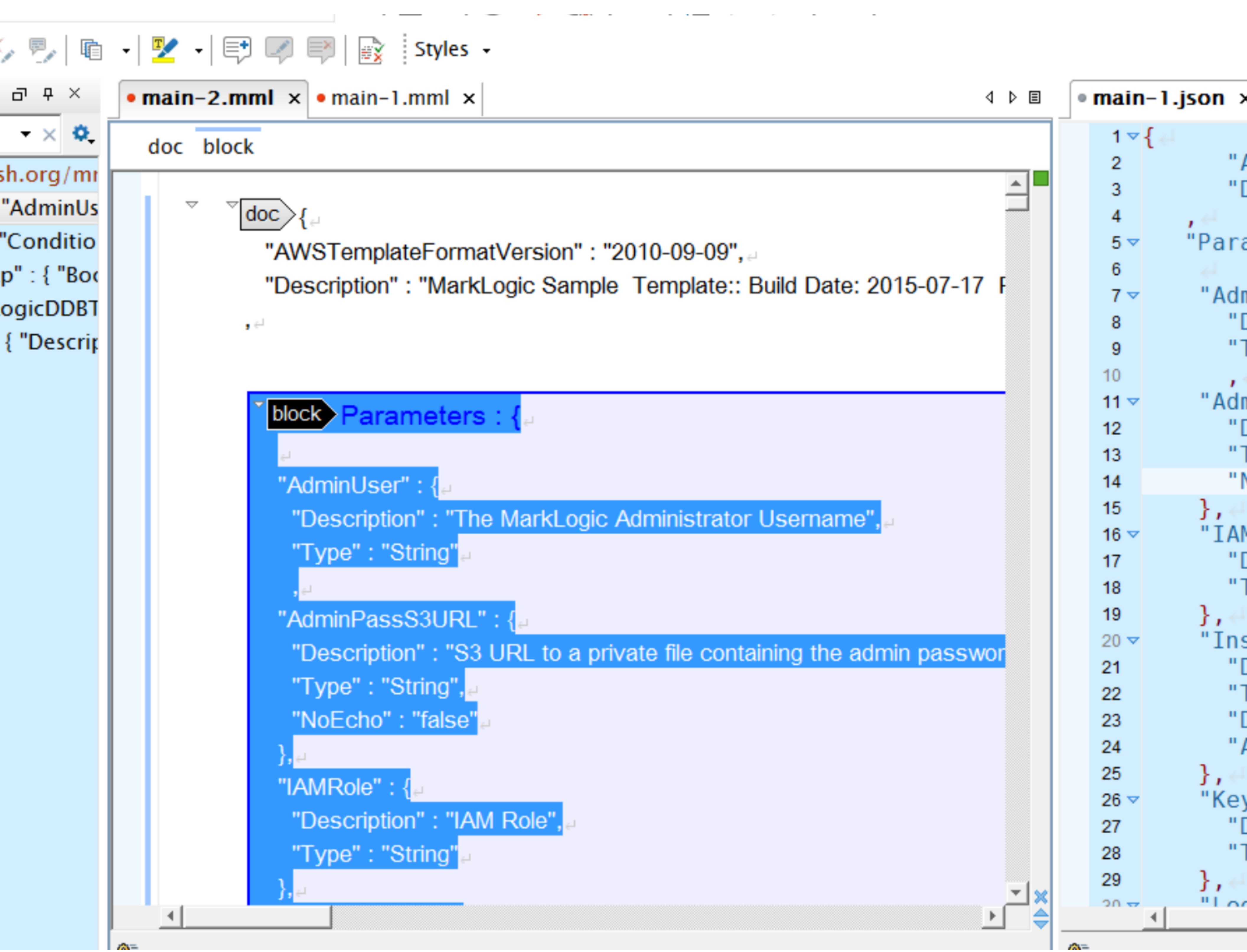
Oxygen Plugin + Extension - + Saxon extensions


main-2.mml [K:\Qsync\MarkLogic\Balisage2015\Presentation\Master\mml-samples\main-2.mml] - <oXygen/> XML Editor
File Edit Find Project Options Tools Menu Document Window Help
markup
XPath 2.0
Execute XPath on 'Current File'
Styles
Outline
Element name filter
main-2.mml
main-1.mml
main-1.json
doc block
Text Grid Author
Text Grid
K:\...\Balisage2015\Presentation\Master\mml-samples\main-2.mml
U+002C
10 new mes...

```
doc "http://www.xmlsh.org/mr
  block "Parameters" "AdminUs
  block "Conditions" "Conditio
  block "json" "VMMap" : { "Boc
  block "json" "MarkLogicDDBT
  block "json" "URL" : { "Descrip
```

```
doc {
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Description" : "MarkLogic Sample  Template:: Build Date: 2015-07-17  F
,

block Parameters : {

  "AdminUser" : {
    "Description" : "The MarkLogic Administrator Username",
    "Type" : "String"
    ,
  "AdminPassS3URL" : {
    "Description" : "S3 URL to a private file containing the admin passwor
    "Type" : "String",
    "NoEcho" : "false"
  },
  "IAMRole" : {
    "Description" : "IAM Role",
    "Type" : "String"
  },
```

```
{
      "AWSTemplateFormatVersion" : "2010-09-09",
      "Description" : "MarkLogic Sample  Template:: Bui
   ,
   "Parameters" : {

    "AdminUser" : {
      "Description" : "The MarkLogic Administrator Usern
      "Type" : "String"
      ,
    "AdminPassS3URL" : {
      "Description" : "S3 URL to a private file contain
      "Type" : "String",
      "NoEcho" : "false"
    },
    "IAMRole" : {
      "Description" : "IAM Role",
      "Type" : "String"
    },
    "InstanceType" : {
      "Description" : "Type of EC2 instance to launch",
      "Type" : "String",
      "Default" : "m1.medium",
      "AllowedValues" : [ "m1.small", "m1.medium", "m1.
    },
    "KeyName" : {
      "Description" : "Name of and existing EC2 KeyPair
      "Type" : "String"
    },
```



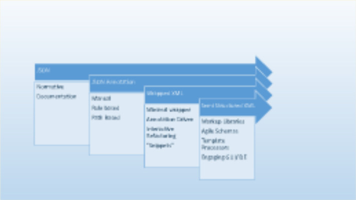
```
doc {
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Description" : "MarkLogic Sample  Template:: Build Date: 2015-07-17  F
,

block Parameters : {

  "AdminUser" : {
    "Description" : "The MarkLogic Administrator Username",
    "Type" : "String"
    ,
  "AdminPassS3URL" : {
    "Description" : "S3 URL to a private file containing the admin passwor
    "Type" : "String",
    "NoEcho" : "false"
  },
  "IAMRole" : {
    "Description" : "IAM Role",
    "Type" : "String"
  },
```

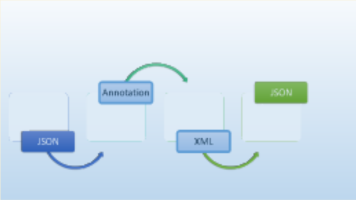
```
sh.org/mr
"AdminUs
"Conditio
p" : { "Boc
ogicDDBT
{ "Descrip
```

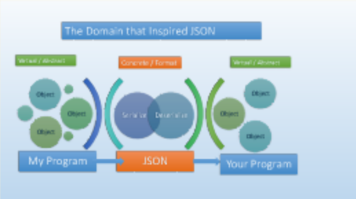
```
{
    "A
    "D
  ,
  "Para

   "Adm
    "D
    "T
    ,
   "Adm
    "D
    "T
    "M
   },
   "IAM
    "D
    "T
   },
   "Ins
    "D
    "T
    "D
    "A
   },
   "Key
    "D
    "T
   },
```

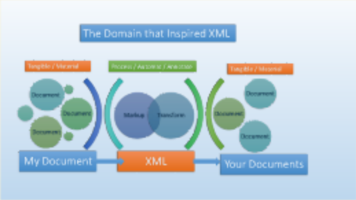
Concept:

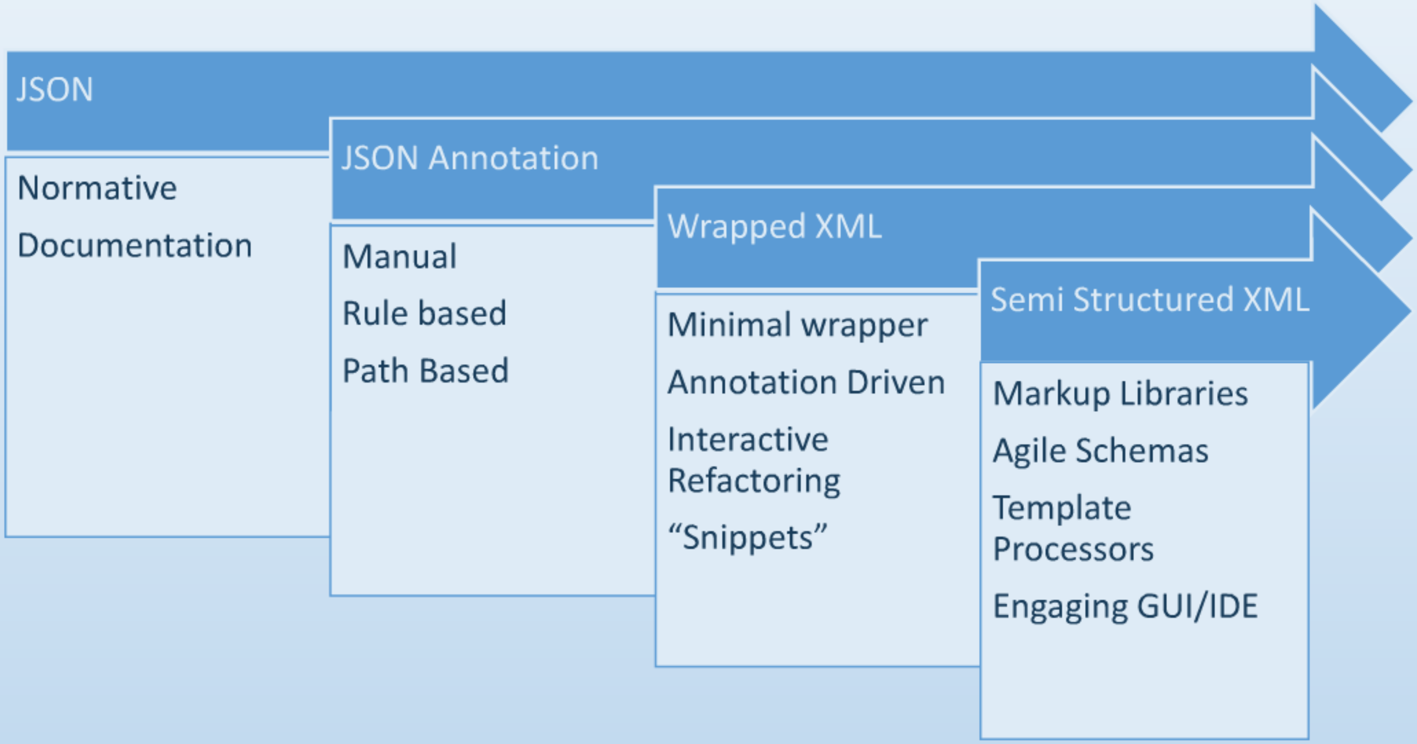
Keep Thinking !!











JSON

Normative

Documentation

JSON Annotation

Manual

Rule based

Path Based

Wrapped XML

Minimal wrapper

Annotation Driven

Interactive Refactoring

“Snippets”

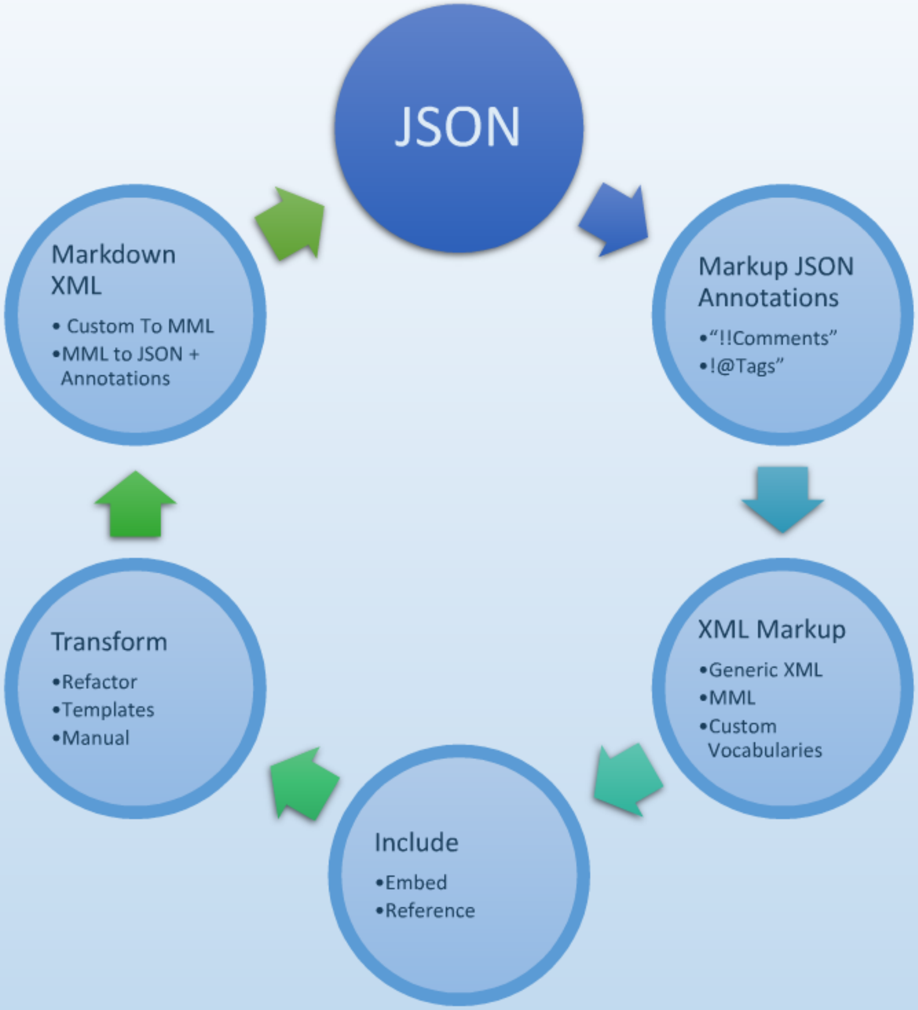
Semi Structured XML

Markup Libraries

Agile Schemas

Template Processors

Engaging GUI/IDE



JSON

Markup JSON Annotations

- “!!Comments”
- !@Tags”

XML Markup

- Generic XML
- MML
- Custom Vocabularies

Include

- Embed
- Reference

Transform

- Refactor
- Templates
- Manual

Markdown XML

- Custom To MML
- MML to JSON + Annotations

So what IS "MML" ??

A "Language" for Multiple Markups ..

"Language"

Communication

Conversation

A way of thinking

Of many views

So what IS "MML" ??

A "Language" for Multiple Markups ..

"Language"

age"

Communication

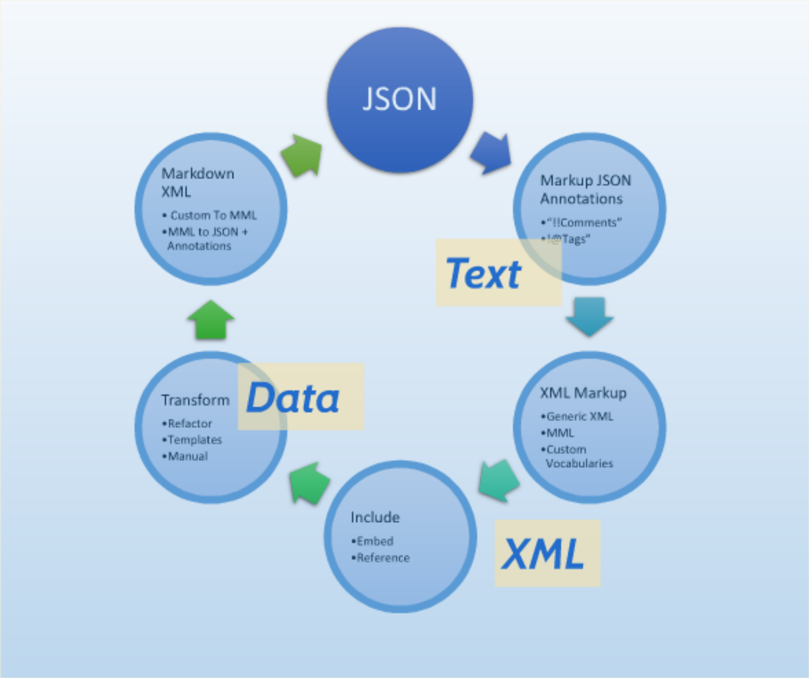
Conversation

on

A way of thinking

Of many views

A superposition of many theories

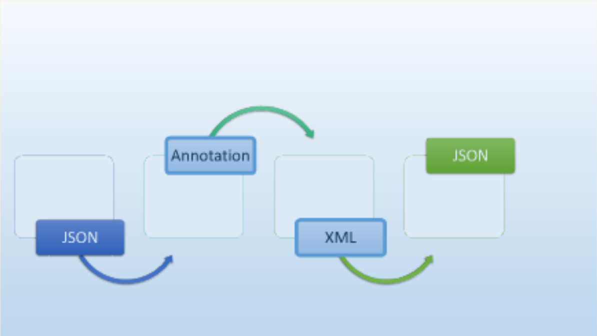


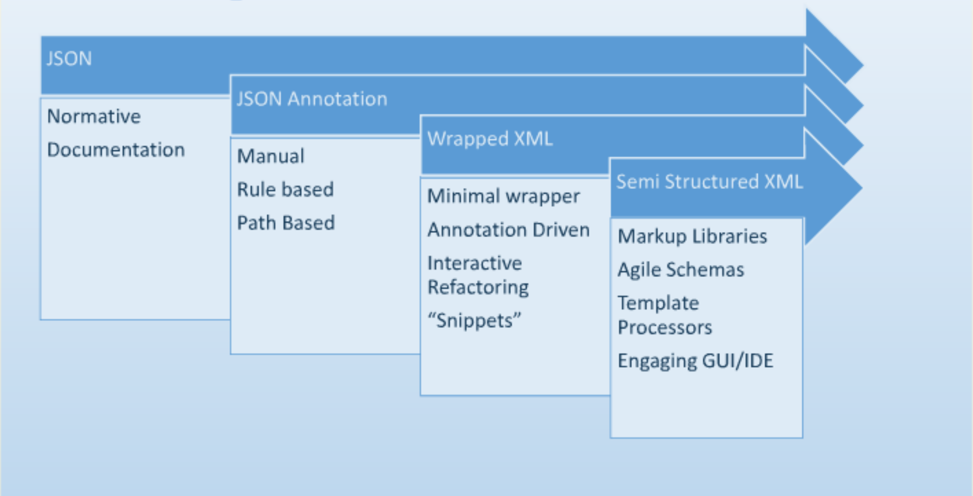
When one just inst enough

How can Markup be dead ?

Its not just a good theory

Continuous Refactoring





JSON

Normative
Documentation

JSON Annotation

Manual
Rule based
Path Based

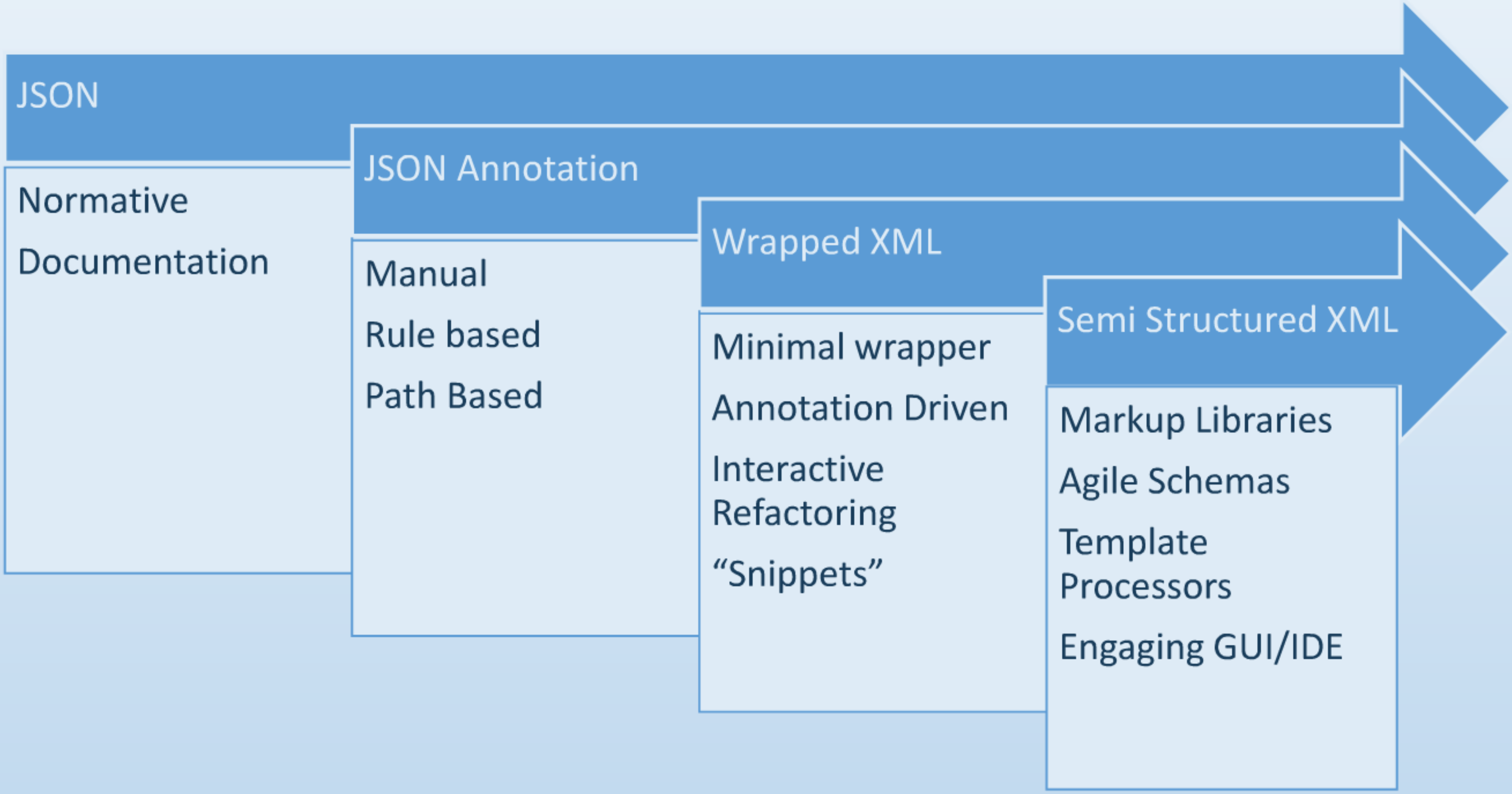
Wrapped XML

Minimal wrapper
Annotation Driven
Interactive Refactoring
"Snippets"

Semi Structured XML

Markup Libraries
Agile Schemas
Template Processors
Engaging GUI/IDE

efactoring



JSON

Normative

Documentation

JSON Annotation

Manual

Rule based

Path Based

Wrapped XML

Minimal wrapper

Annotation Driven

Interactive Refactoring

“Snippets”

Semi Structured XML

Markup Libraries

Agile Schemas

Template Processors

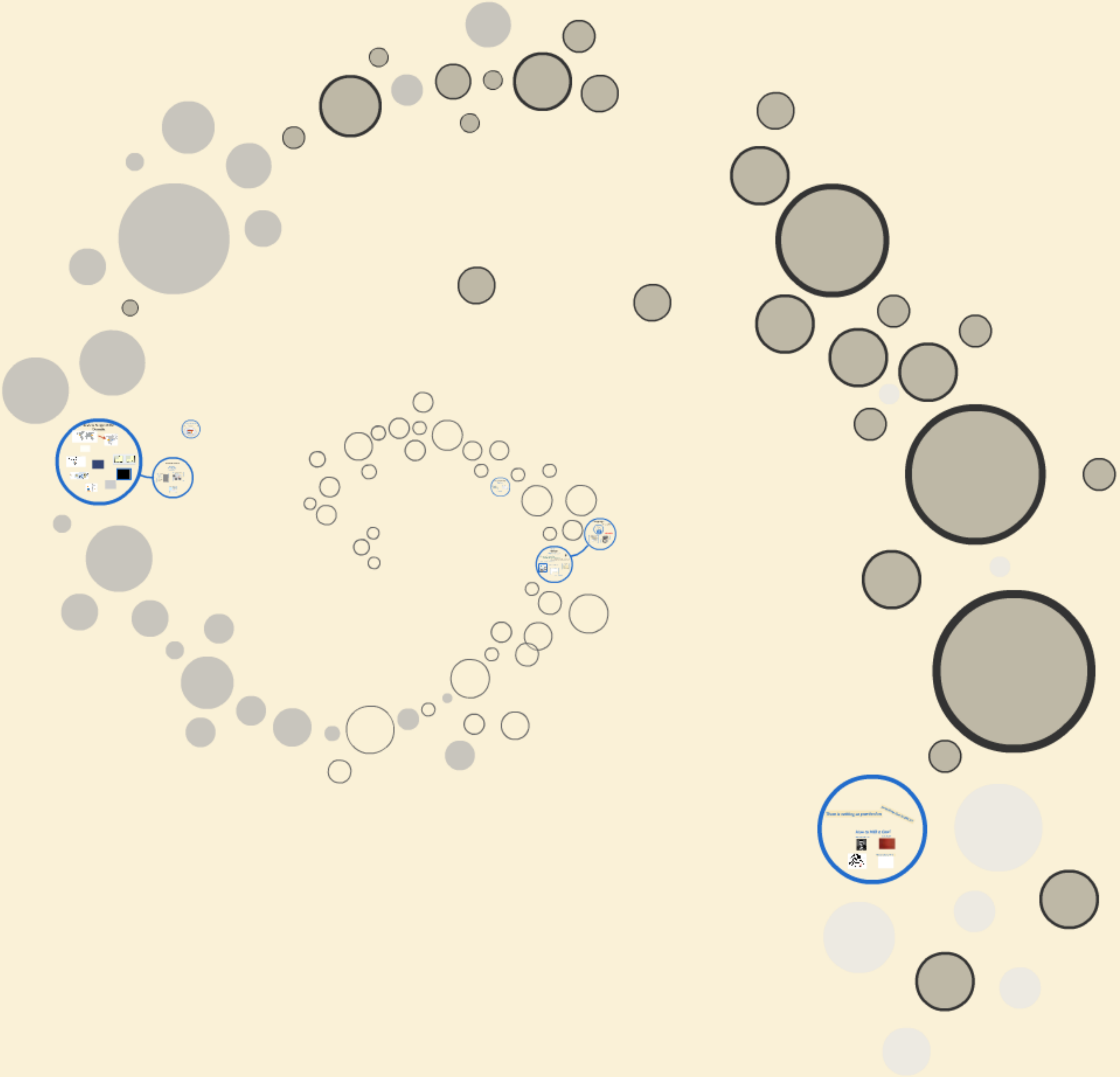
Engaging GUI/IDE

When one just inst enough

How can Markup be dead ?

Its not just a good theory

Its not just a good theory


There is nothing as powerful as
How to Kill a Deer!