

Network Configuration Management Via Model Finding

Sanjai Narain
Senior Research Scientist
Telcordia Technologies
narain@research.telcordia.com

Large, complex distributed systems are created via configuration

- Every component has finite number of configuration parameters. Each is set to a definite value to satisfy systemwide requirements
- System wide requirements are on e.g., functionality, security, fault-tolerance, performance
- Configuration is “machine language” for logical, system integration
- Relevance to self-managing systems:
 - Logically integrate self-managing systems into larger ones
 - Dynamically reconfigure systems to satisfy systemwide requirements

Yet, there is no “theory” of configuration

System Requirements

Configuration
Synthesis

Requirement
Strengthening

Component Adds &
Deletes

Configuration Error
Diagnosis

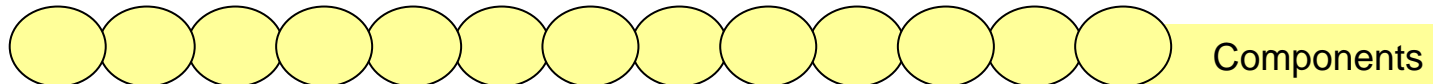
Configuration Error
Fixing

Requirement
Verification

These **reasoning** tasks are all manually performed

System requirements can't even be precisely specified, hence automation of reasoning tasks is impossible

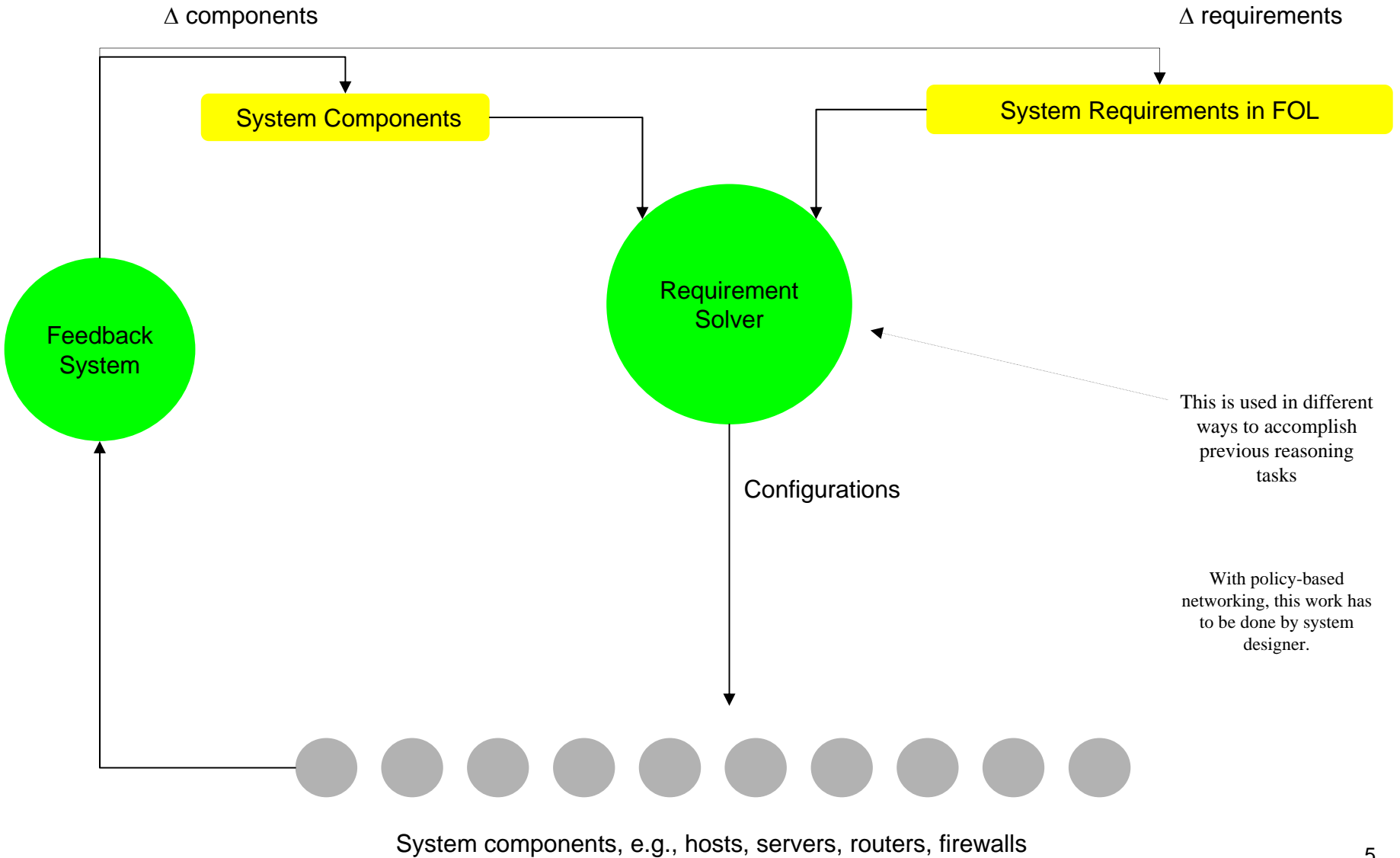
Leads to high cost of infrastructure ownership



Designing Requirements Language

- Semantic aspect: What are intuitive abstractions (logical structures, relationships) used by system administrators?
 - FSM models of protocols are impractical
- Syntactic aspect: How to combine abstractions into requirements?
 - Propositional logic, definite clauses, FOL, higher-order logic, temporal logic?
- Progress to date: “Service Grammar”:
 - Semantic aspect: Formalize notion of “correct configuration” associated with protocols.
 - Syntactic aspect: Definite clauses
 - “Building Autonomic Systems via Configuration”, Proceedings of Autonomic Systems Workshop, 2003
- However, FOL is often required
 - But theorem provers have not been very efficient
 -until now, with advent of SAT solvers

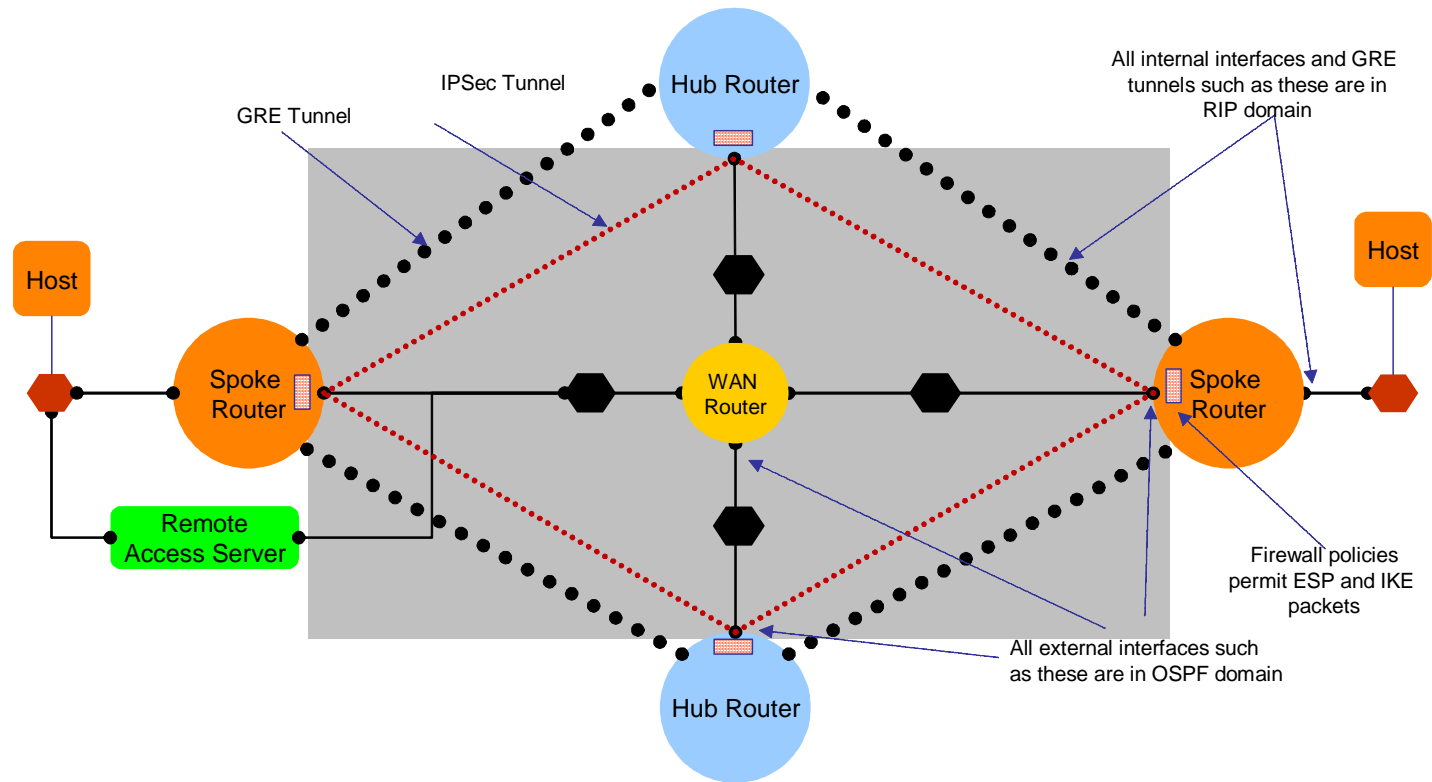
New Concept: Requirement Solver



Implementation in Alloy

- Developed by Professor Daniel Jackson's group at MIT
- Allows specification of:
 - Objects types, parameters and value types
 - First-order logic constraints on values
 - Scope: number and type of each object
- Given a specification, Alloy tries to find its “model”, i.e., assignment of parameters to values to satisfy constraints
- Compiles specification into Boolean formula then uses SAT solvers

Fault-Tolerant VPN (Overlay)



Phase II: Create several VPNs, one for each level of sensitivity
Phase III: Merge collections of mobile VPNs

Current VPN Configuration Process

New Cisco IOS configuration needs to be implemented at all VPN peer routers! For 4 node VPN that is more than 240 command lines.

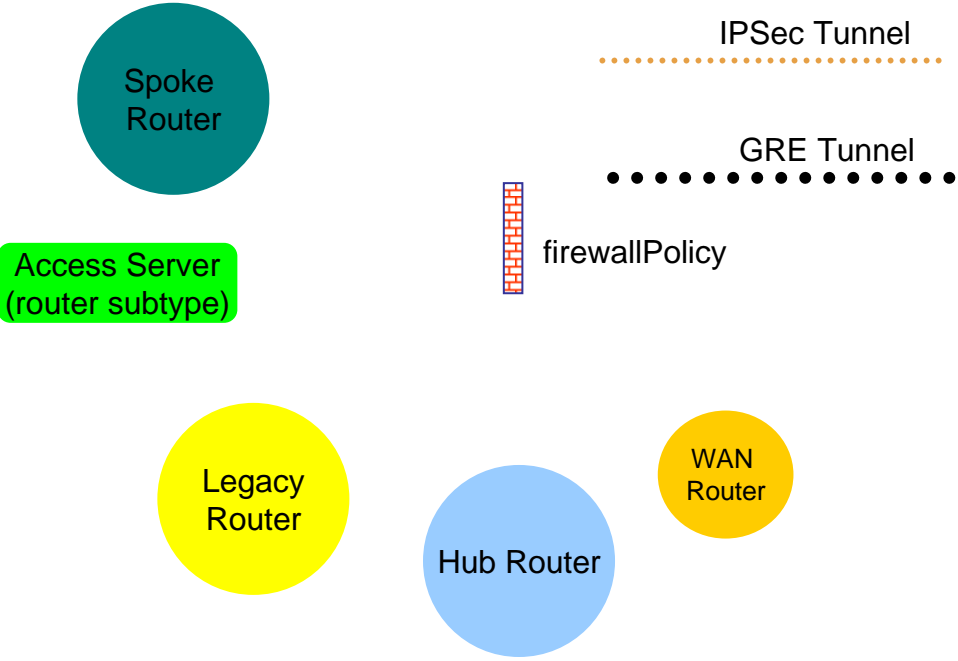
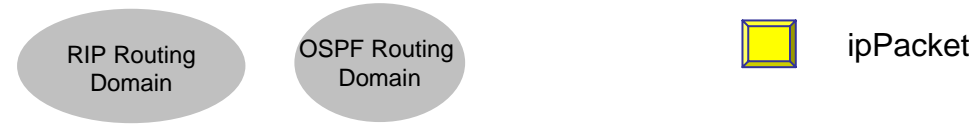
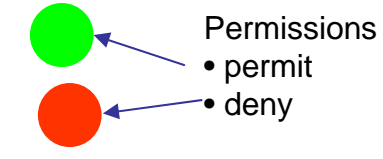
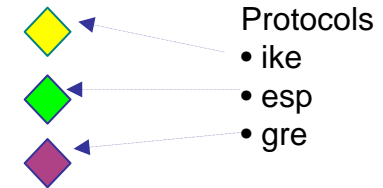
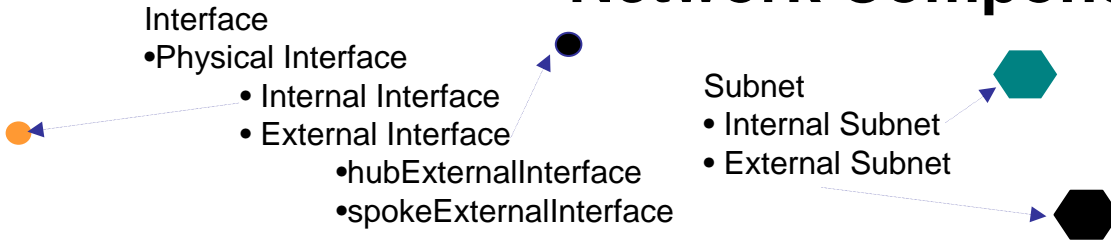
Realistic deployment:

- 240 sites
- Can take years
- VPN services market in 2003: \$18 billion

```
hostname AI-RTR
!
interface Ethernet0/0
 ip address 158.158.158.2
 tunnel source 158.158.158.2
 tunnel destination 128.128.128.2
 crypto map vpn-map-Ethernet0/0
!
interface Tunnel0
 ip address 35.35.35.2 255.255.255.0
!
interface Tunnel1
 ip address 33.33.33.2 255.255.255.0
 tunnel source 158.158.158.2
 tunnel destination 148.148.148.2
 crypto map vpn-map-Ethernet0/0
!
end

hostname AI-RTR
!
crypto isakmp policy 1
 authentication pre-share
 crypto isakmp key SN1BS-RTR_key_with_AI-RTR address 128.128.128.2
 crypto isakmp key PN1BS-RTR_key_with_AI-RTR address 148.148.148.2
 crypto isakmp key SN2-RTR_key_with_AI-RTR address 138.138.138.2
!
crypto ipsec transform-set IPsecProposal esp-des esp-sha-hmac
!
crypto map vpn-map-Ethernet0/0 33 ipsec-isakmp
 set peer 128.128.128.2
 set transform-set IPsecProposal
 match address 142
!
crypto map vpn-map-Ethernet0/0 34 ipsec-isakmp
 set peer 148.148.148.2
 set transform-set IPsecProposal
 match address 143
!
router rip
 version 2
 network 35.35.35.0
 set peer 128.128.128.2
 set transform-set IPsecProposal
 match address 144
!
interface Tunnel0
 ip address 35.35.35.2 255.255.255.0
 tunnel source 158.158.158.2
 tunnel destination 128.128.128.2
 crypto map vpn-map-Ethernet0/0
!
interface Tunnel1
 ip address 33.33.33.2 255.255.255.0
 tunnel source 158.158.158.2
 tunnel destination 148.148.148.2
 crypto map vpn-map-Ethernet0/0
!
end
```

Network Components



Component Attributes

- **interface**
 - chassis: router
 - network: subnet
 - routing: routingDomain
- **ipsecTunnel**
 - local: externalInterface,
 - remote: externalInterface,
 - protocolToSecure: protocol
- **greTunnel**
 - localPhysical: externalInterface
 - remotePhysical: externalInterface
 - routing: routingDomain
- **firewallPolicy**
 - prot: protocol
 - action: permission
 - protectedInterface: physicalInterface
- **ipPacket**
 - source: interface,
 - destination: interface,
 - prot: protocol

List of Network Requirements

RouterInterfaceRequirements

1. Each spoke router has internal and external interfaces
2. Each access server has internal and external interfaces
3. Each hub router has only external interfaces
4. Each WAN router has only external interfaces

SubnettingRequirements

5. A router does not have more than one interface on a subnet
6. All internal interfaces are on internal subnets
7. All external interfaces are on external subnets
8. Every hub and spoke router is connected to a WAN router
9. No two non-WAN routers share a subnet

RoutingRequirements

10. RIP is enabled on all internal interfaces
11. OSPF is enabled on all external interfaces

GRERequirements

12. There is a GRE tunnel between each hub and spoke router
13. RIP is enabled on all GRE interfaces

SecureGRERequirements

14. For every GRE tunnel there is an IPSec tunnel between associated physical interfaces that secures all GRE traffic

AccessServerRequirements

15. There exists an access server and spoke router such that the server is attached in "parallel" to the router

FirewallPolicyRequirements

16. Each hub and spoke external interface permits esp and ike packets

Human administrators reason with these in different ways to synthesize initial network, then reconfigure it as operating conditions change.

Can we automate this reasoning?

Strengthening Requirement: Adding Overlay Network

RouterInterfaceRequirements

1. Each spoke router has internal and external interfaces
2. Each access server has internal and external interfaces
3. Each hub router has only external interfaces
4. Each WAN router has only external interfaces

SubnettingRequirements

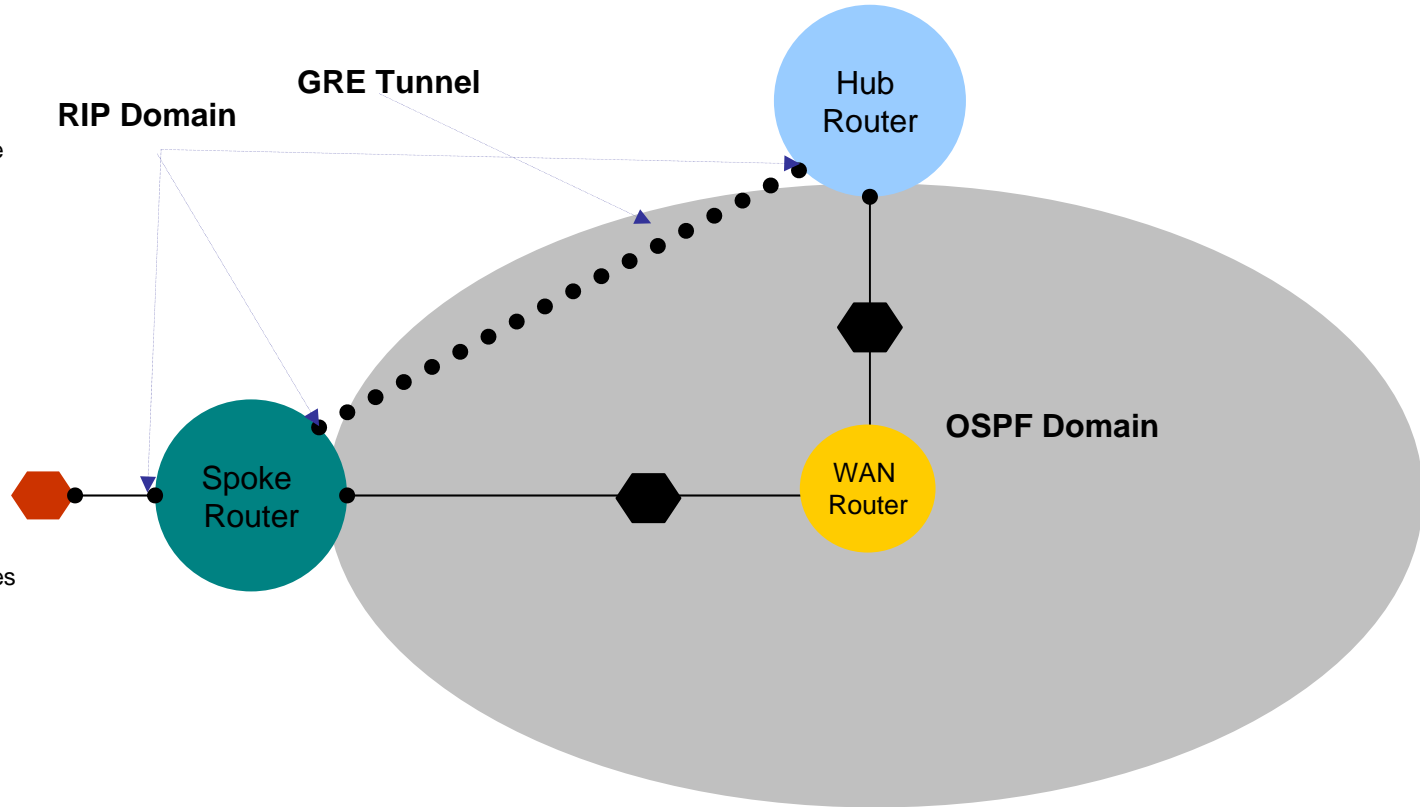
5. A router does not have more than one interface on a subnet
6. All internal interfaces are on internal subnets
7. All external interfaces are on external subnets
8. Every hub and spoke router is connected to a WAN router
9. No two non-WAN routers share a subnet

RoutingRequirements

10. RIP is enabled on all internal interfaces
11. OSPF is enabled on all external interfaces

GRERequirements

12. There is a GRE tunnel between each hub and spoke router
13. RIP is enabled on all GRE interfaces



To synthesize network, satisfy R1-R13 for

- previous list of components &
- 1 GRE tunnel

NOTE: GRE tunnel set up and RIP domain extended to include GRE interfaces automatically!

Strengthening Requirement: Adding Security For Overlay Network

RouterInterfaceRequirements

1. Each spoke router has internal and external interfaces
2. Each access server has internal and external interfaces
3. Each hub router has only external interfaces
4. Each WAN router has only external interfaces

SubnettingRequirements

5. A router does not have more than one interface on a subnet
6. All internal interfaces are on internal subnets
7. All external interfaces are on external subnets
8. Every hub and spoke router is connected to a WAN router
9. No two non-WAN routers share a subnet

RoutingRequirements

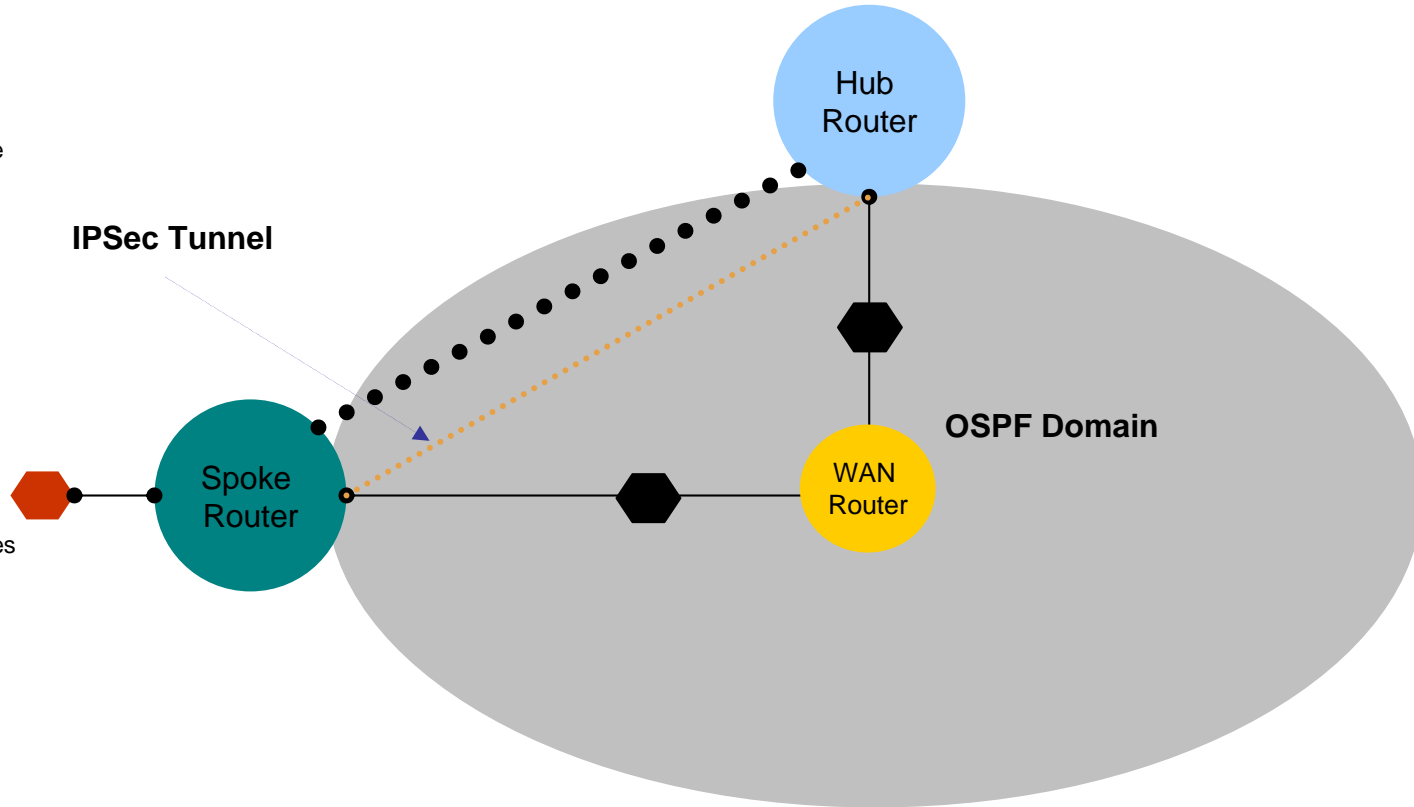
10. RIP is enabled on all internal interfaces
11. OSPF is enabled on all external interfaces

GRERequirements

12. There is a GRE tunnel between each hub and spoke router
13. RIP is enabled on all GRE interfaces

SecureGRERequirements

14. For every GRE tunnel there is an IPSec tunnel between associated physical interfaces that secures all GRE traffic



To synthesize network, satisfy R1-R14 for

- previous list of components &
- 1 IPSec tunnel

NOTE: IPSec tunnel securing GRE tunnel set up automatically

Strengthening Requirement: Adding Remote Access Service

RouterInterfaceRequirements

1. Each spoke router has internal and external interfaces
2. Each access server has internal and external interfaces
3. Each hub router has only external interfaces
4. Each WAN router has only external interfaces

AccessServerRequirements

15. There exists an access server and spoke router such that the server is attached in "parallel" to the router

SubnettingRequirements

5. A router does not have more than one interface on a subnet
6. All internal interfaces are on internal subnets
7. All external interfaces are on external subnets
8. Every hub and spoke router is connected to a WAN router
9. No two non-WAN routers share a subnet

RoutingRequirements

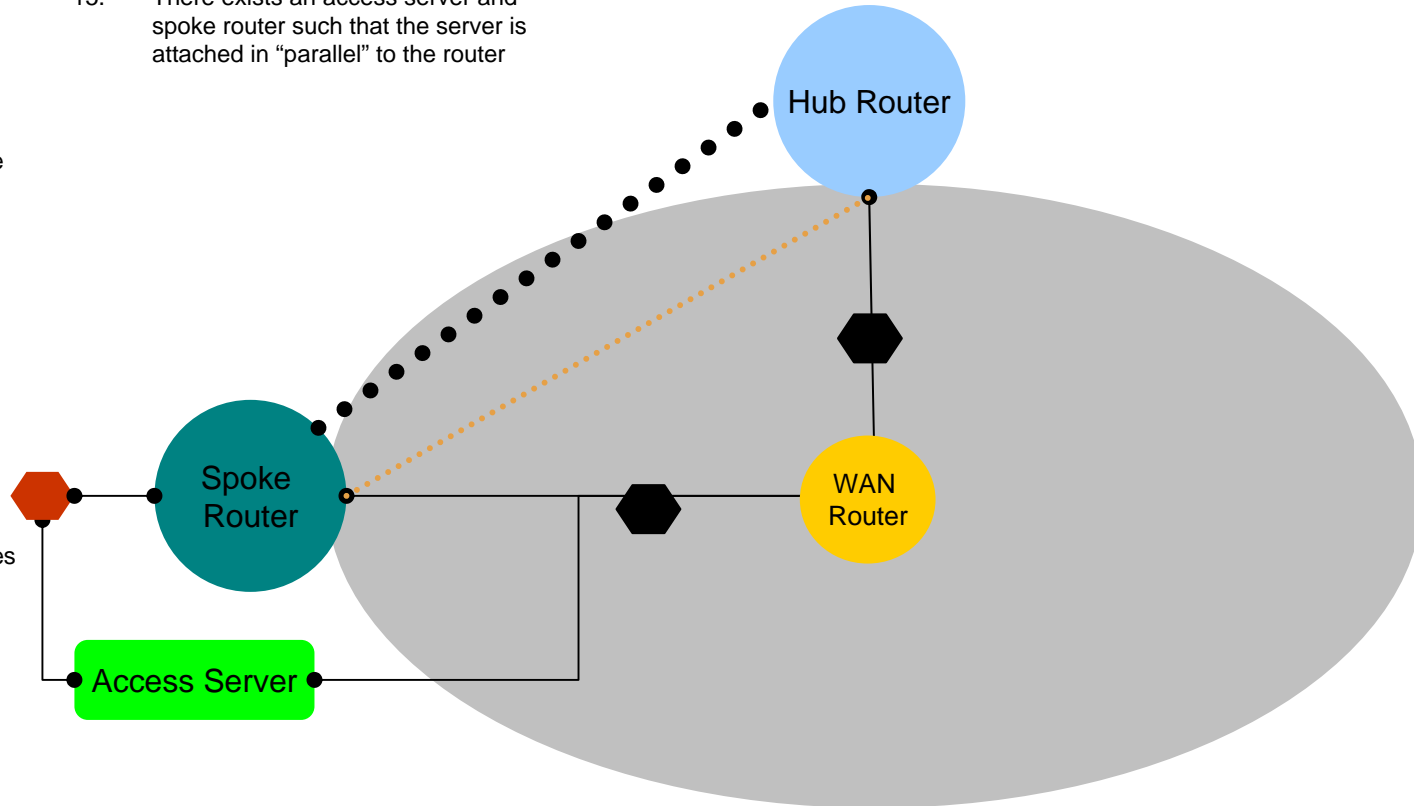
10. RIP is enabled on all internal interfaces
11. OSPF is enabled on all external interfaces

GRERequirements

12. There is a GRE tunnel between each hub and spoke router
13. RIP is enabled on all GRE interfaces

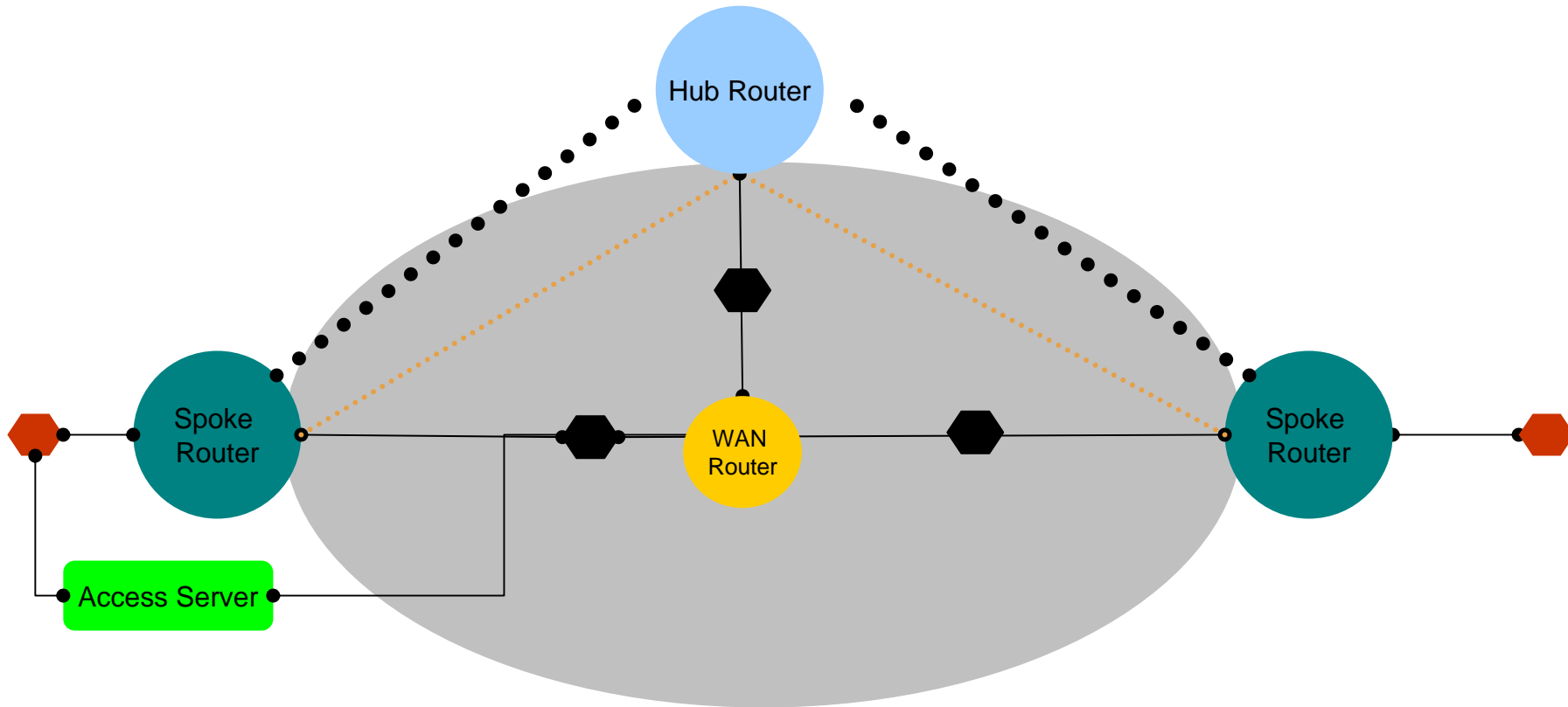
SecureGRERequirements

14. For every GRE tunnel there is an IPSec tunnel between associated physical interfaces that secures all GRE traffic



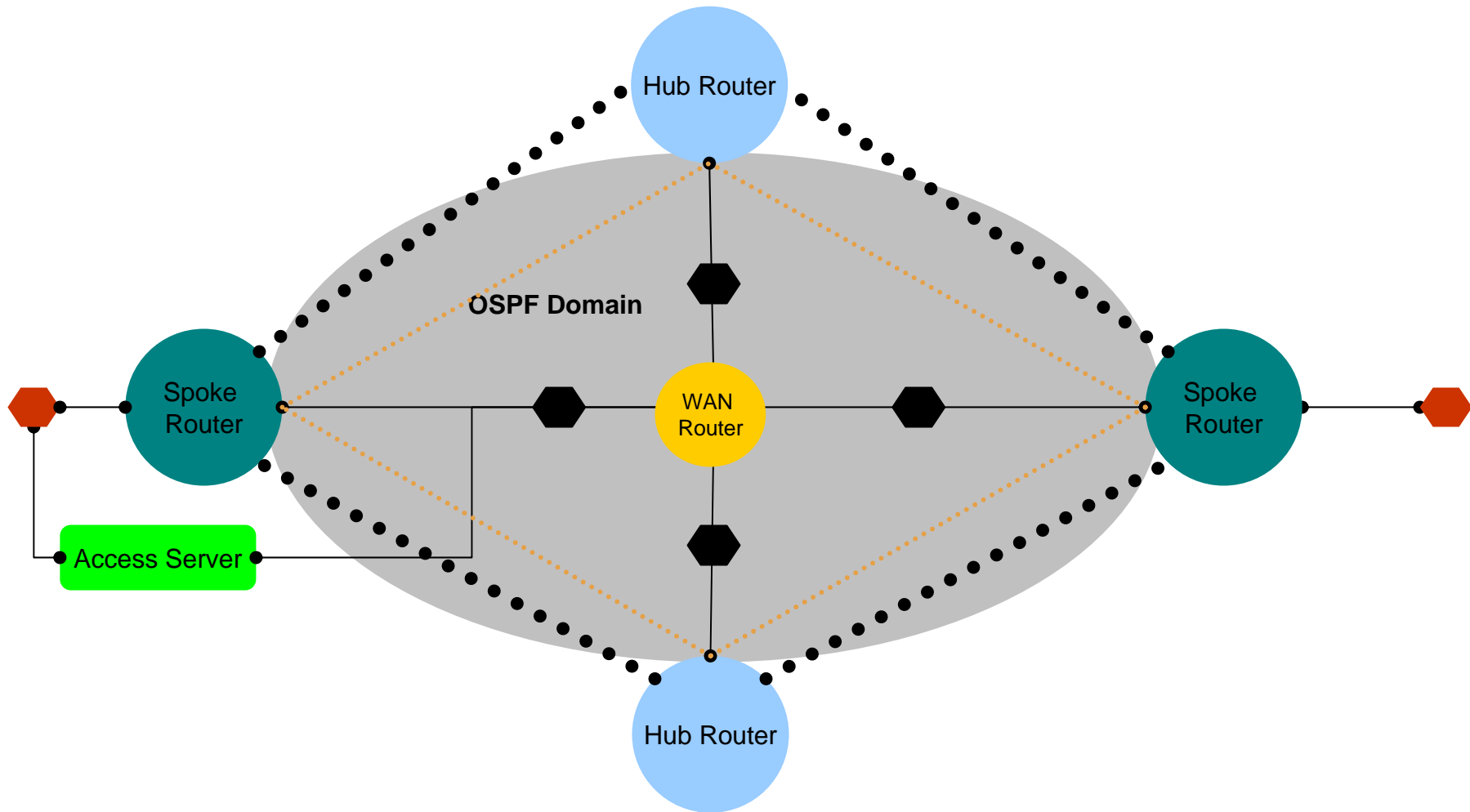
- To synthesize network, satisfy R1-R15 for previous list of components and 1 additional access server.
- Note: Access server interfaces placed on correct interfaces and RIP and OSPF domains correctly extended with internal and external interfaces, respectively

Component Addition: Adding New Spoke Router



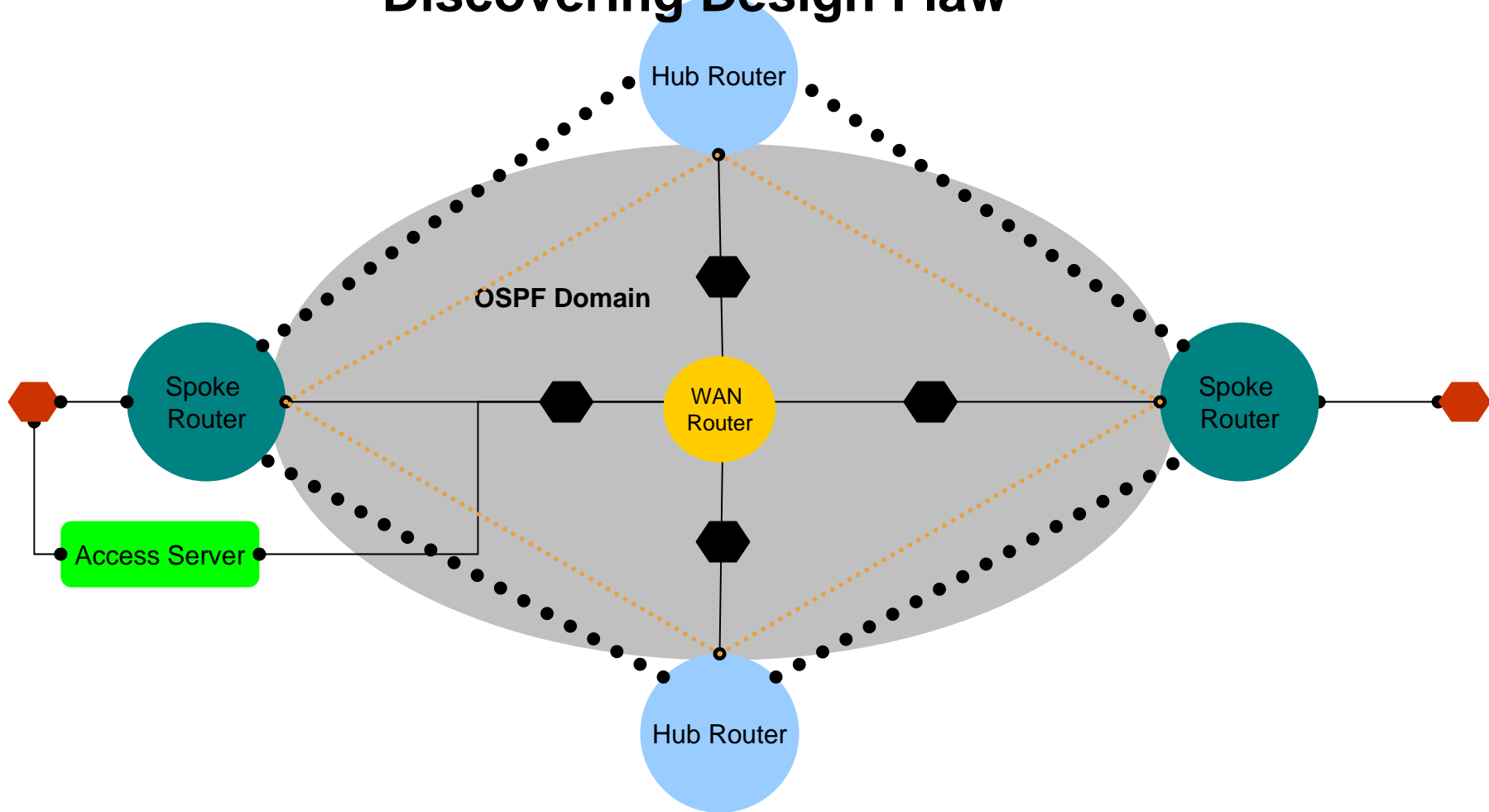
- To add another spoke router satisfy requirements R1-R16 for previous components and one additional spoke router and related components
- Note: New subnets, GRE and IPsec tunnels set up, and routing domains extended *automatically*

Component Addition: Adding New Hub Router



- To add another hub router satisfy requirements R1-R16 for previous components and one additional hub router (and related components)
- New subnets, GRE and IPSec tunnels set up, and routing domains extended *automatically*

Verification: Adding Firewall Requirements & Discovering Design Flaw



- Symptom: Cannot ping from one internal interface to another
- Define Bad = ip packet is blocked
- Check if R1-R16 & Bad is satisfiable
- Answer: WAN router firewalls block ike/ipsec traffic
- Action: Create new policy that allows WAN router firewalls to pass esp/ike packets

Summary And Future Directions

- Summary
 - Proposed a theory of configuration
 - Designed requirements language + reasoning operations
 - Developed strategies for “efficient specification”
 - Showed implementation in Alloy in context of realistic VPN
- Future directions
 - Close the loop to create self-managing systems
 - Incremental configuration
 - Scalability to thousands of nodes (efficient specification)
 - Distributed constraint solvers
 - Distributed self-management

