ALLSEEN ALLIANCE

Technical Steering Meeting

Aug 11, 2014

Antitrust Compliance Notice

- AllSeen Alliance meetings involve participation by industry competitors, and it is the intention of AllSeen Alliance to conduct all of its activities in accordance with applicable antitrust and competition laws. It is therefore extremely important that attendees adhere to meeting agendas, and be aware of and not participate in any activities that are prohibited under applicable US state, federal or foreign antitrust and competition laws.

- Examples of types of actions that are prohibited at AllSeen Alliance meetings and in connection with AllSeen Alliance activities are described in the AllSeen Alliance Antitrust Policy. If you have questions about these matters, please contact your company counsel, or if you are a member of AllSeen Alliance, feel free to contact Lee Gesmer or Andrew Updegrove, of the firm of Gesmer Updegrove LLP, which provides legal counsel to AllSeen Alliance.

Reminder:

This call is being recorded

3

Agenda

1. Approve minutes from previous meeting
2. Release process evolution - proposal
3. AllJoyn Discovery Redesign (14.02 to 14.06)

Proposal for Evolving the Release Process

Introduction:
Lessons learned over the last few releases

- Coordinating a large number of features
  - A significant factor in the delay of 14.06 release
- Need better definition of quality level
  - Not possible for one member to ensure release works on all member's products
  - Need to define a process for members to contribute
    - Platform specific support
    - Release testing
- Working Group coordination issues
  - Coordination between two working groups (base services and core) was challenging
  - As more WGs are added, this will become more difficult

Proposed changes

- Separate features from releases
  - Features are targeted for releases but releases do not depend on specific features
  - Features are developed and tested on public branches
    - If a feature does not meet quality for the targeted release, it is slipped to the following release
  - Once at release quality features are merged into the master branch for inclusion into next release
    - Release quality implies feature testing is complete
- Members are responsible for integration testing and hardening on their products
  - Need member input on process for platform specific contributions and testing
- Move away from a simultaneous release model
  - Each Working Group (WG) determines when they will release
  - Core currently on a 4 month cadence (February, June, & October)
    - Core WG would like TSC approval to change to a 6 month cadence (April & October)
  - Cross WG integration and resolution of technical dependencies is coordinated by the working groups

Benefits / details of each recommendation

Separate features from releases

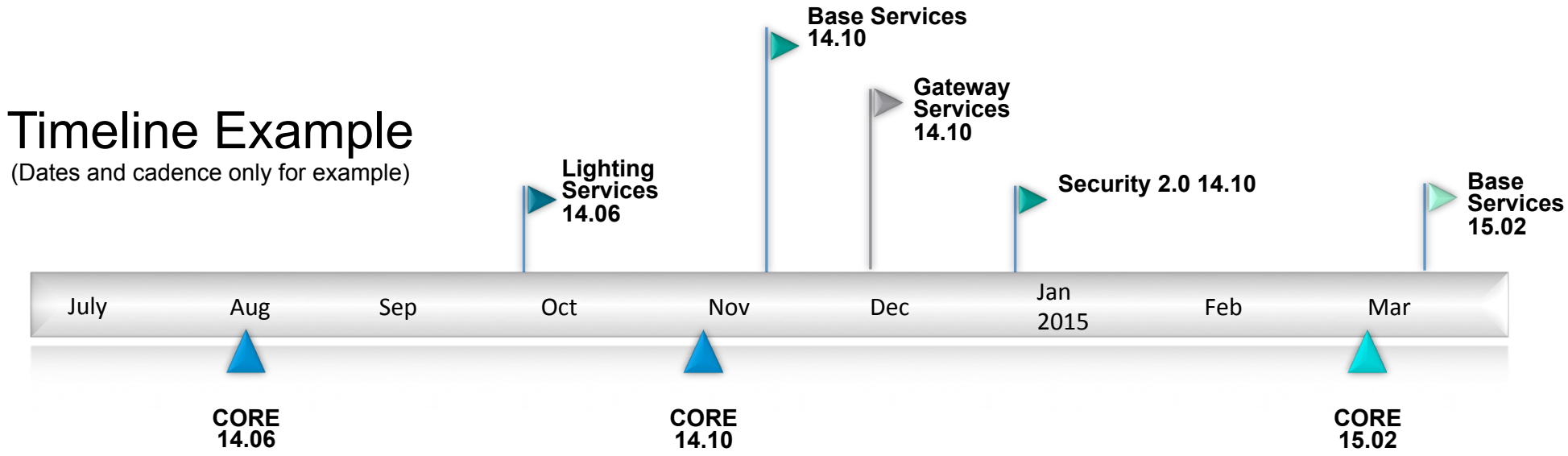
- Allows members to take features that have not been released
  - Members can harden the feature branch and contribute bug fixes, etc. to the WG
- Allows working groups to make better trade off decisions on feature priorities
  - If a feature becomes high priority, WG can better understand the impact on the lower priority features without having to contend with the “global” view
- Allows features that do not meet the quality bar to be slipped without impacting the release date
- A Release is defined as an effort to validate on a specific set of platforms
  - Each working group defines the platforms supported and the timing of their release
  - The name/number of the release follows the core version on which it is based
    - Example on slide 11

Members are responsible for integration testing and hardening on their products

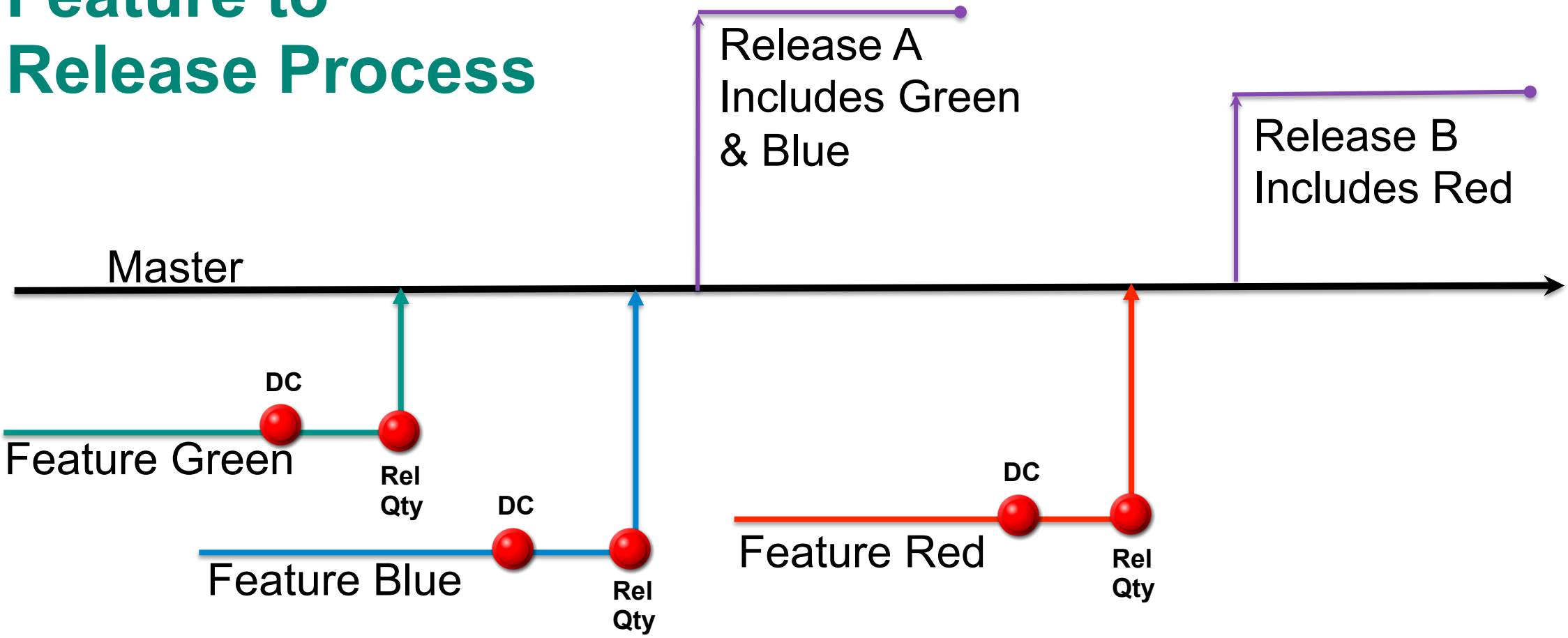
- Each member that “hardens” the code is strongly encouraged for feeding the fixes back to the Alliance code base
  - Not doing so implies the member will have to maintain their own fork
- Need to define how members contribute platform support and testing

Move away from a simultaneous release model

- Working Group releases are based on a specific Core release version
  - Version naming is same as Core release version used
  - WG will identify the number of commits taken in after the Core release
- Examples:
  - Base Services WG commits to having a release for each core version
  - Lighting WG may chose not to update after 14.06
    - This implies that there would only be a Lighting 14.10 if a member chooses to do the work



Feature to Release Process



DC: Development Complete (All features implemented and only bug fixing remains)
RQ: Release Quality (All feature testing is complete and feature is merged)
Releases: Focused on feature integration, system, and regression testing

Thank You

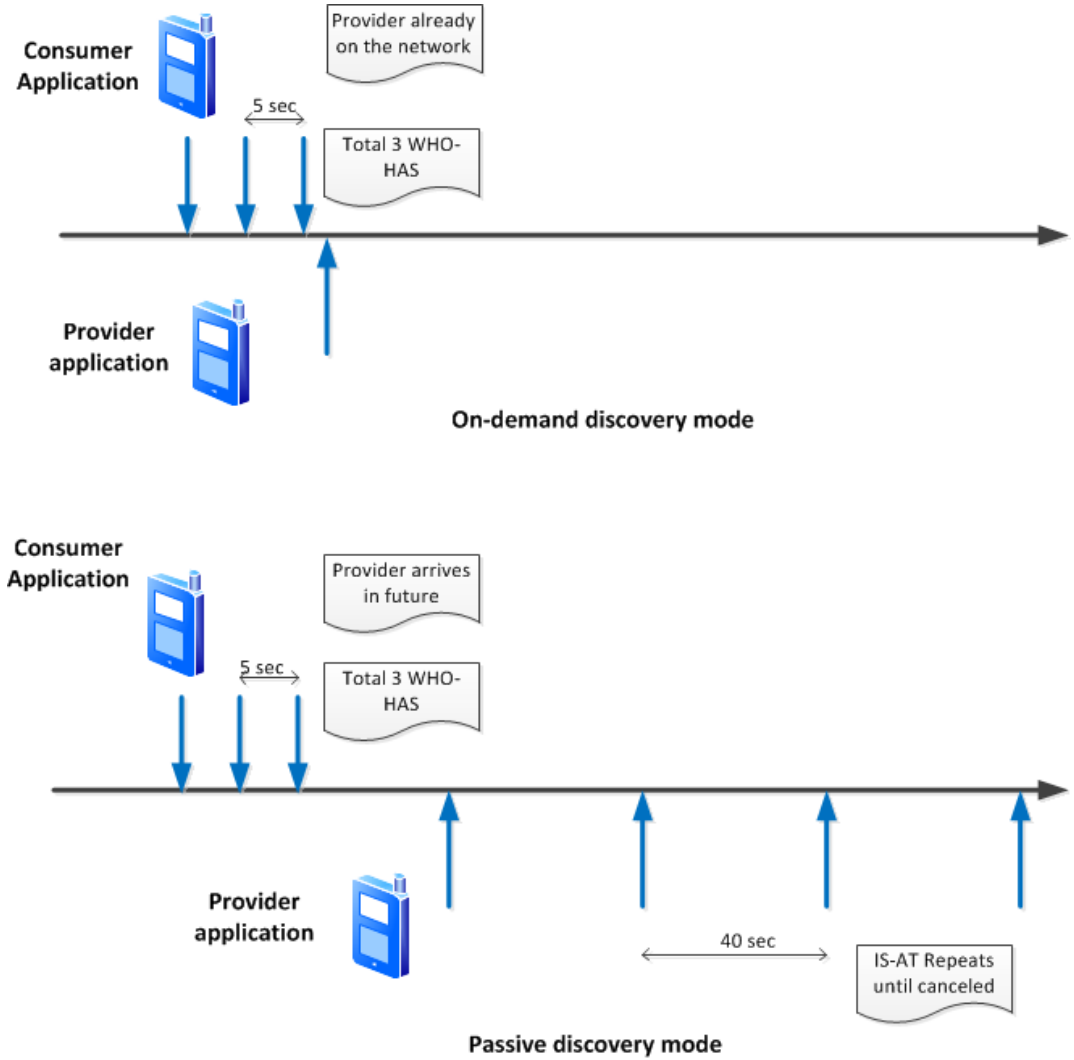
- Proposal review and comment period
  - 8/5/14 to 8/18/14
  - Members encouraged to use the TSC mailing list if they wish to ask questions and/or comment
- Proposal vote
  - 8/18/14

AllJoyn Discovery Redesign (14.02 to 14.06)

Outline

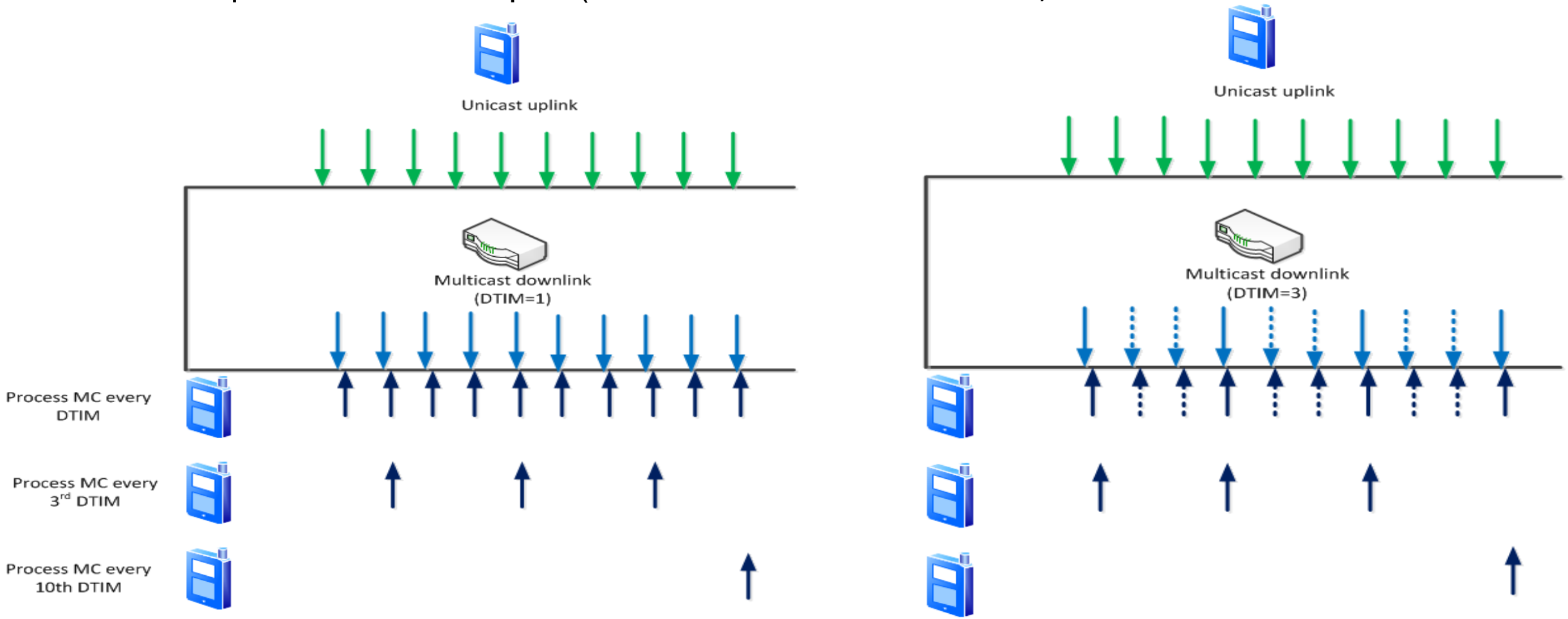
- Review 14.02 (NS) Baseline
  - Wi-Fi Multicast Dependency
  - Analytical model
- Outline Performance Objectives
  - Discovery
  - Presence
- NGNS Results
  - Analytical Model
  - Test Results
- NGNS Stack and APIs
- NGNS Message Sequences

NS Discovery Modes



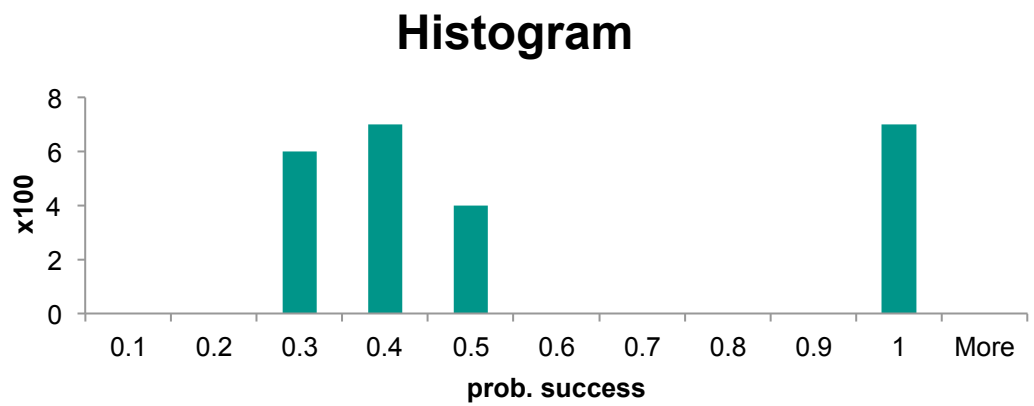
Multicast Reception

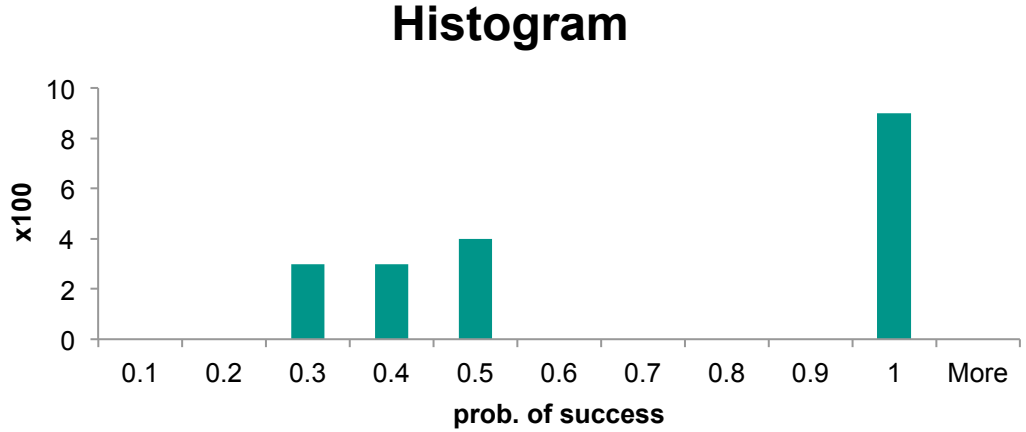
- Send MC packet 100msec apart (beacon interval = 102.4 msec)



Characterization: Multicast over Wi-Fi

- 100% network available; AP DTIM=1; partial wake lock (standby)

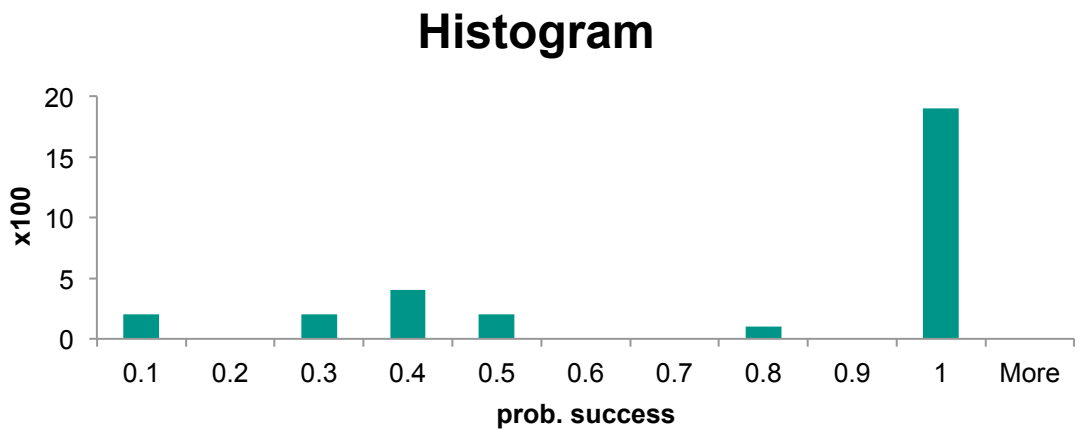




Notice probability of success concentrated around **1/3** or close to 100%.

Characterization: Multicast over Wi-Fi

- 100% network available; AP DTIM=3; partial wake lock (standby)



Notice probability of success concentrated around 1/10, **1/3** or close to 100%.

Latency – Analytical Modeling

- On-demand discovery

Event	Probability of success ()	Discovery Time ()
IS-AT received at first attempt		
IS-AT received at the second attempt		+
IS-AT received at the third attempt		+
IS-AT received after three attempts	1-(+)	

- Passive discovery:

  IS-AT reception can be modeled as Geometric distribution, hence:

  E[Delay] = $q/p\ T\downarrow is + T\downarrow s$

Parameter	Description	Value
p	Probability of successful receipt on multicast	0.35, 0.95
T_{who}	*Timer for WHO-HAS transmissions*	5 sec
T_{is}	*Timer for IS-AT transmissions*	40 sec
T_s	*Discovery time in a successful case*	500 ms

Expected Performance

- Current Baseline - Analytical

Key metrics	**Value (p =0.35)**	**Value (p =0.95)**
Prob. that IS-AT received within first iteration	12.3%	90.3%
Prob. that IS-AT received within three WHO-HAS iterations	32.4%	99.9%
Expected Latency (passive mode)	74.29 sec*	2.11sec
Expected Latency to detect presence (deterministic)	120 sec	120 sec

* Notification benchmarking data supports this estimated result

NGNS Problem Statement

- Discovery:
  - The expected time taken to find a match for both on-demand and passive discovery modes are unacceptable
  - Objectives
    - On demand discovery 50th percentile: 1 sec, 95th percentile: 10 sec
- Presence:
  - The expected time taken to find a name has left the network is unacceptable
  - Objectives
    - On demand discovery 50th percentile: 0.5 sec, 95th percentile: 1 sec

Discovery Use Cases

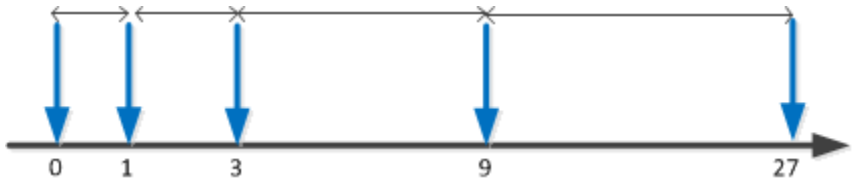
- Make Consumer Application the “Driver”

	Use Case	Realized By
1	Consumer application initiated search	DNS-SD query over *mDNS; mDNS unicast response
2	Provider application initiated unsolicited advertisement	DNS-SD response over mDNS
3	Consumer application polling for the provider application’s existence	Unicast query and Unicast response

*mDNS IP address/port preferred over AllJoyn assigned address/port

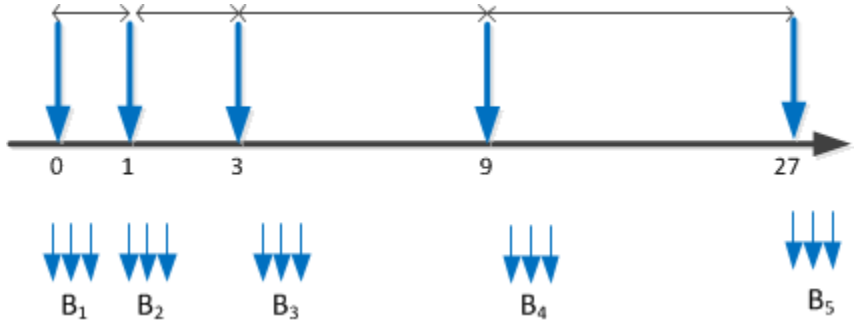
Improved Algorithm

- Achieve 50th percentile: 1sec, 95th percentile: 10sec;
  - Model similarly to Bonjour (Tx = 0,1,3,9)
  - $P=1-(1-p)\uparrow 4$; *p is prob. of success for each transmission*
  - P of 0.95 → p ~ 0.6
  - Determine #quick-repeats to realize p ~ 0.6



Transmission Schedule

- Each MC message follows the transmission schedule
- Each message quick-repeated to address misbehaving devices
- Provides the desired prob. of success (p)
- MC messages within a burst are 100ms apart so that DTIM in successive Beacon frames is set



Analytical Modeling

- New Transmission Schedule + Quick Repeats + Unicast response

Key metrics	**14.02 Value (*p =0.35)**	**14.06 Value (p =0.35)**
Prob. that discovery succeeds within 1 sec	12.3%	60%
Prob. that discovery succeeds within 10 sec	32.4%	97.4%

*p is probability of success of receiving multicast packet

Comparison with Test Data

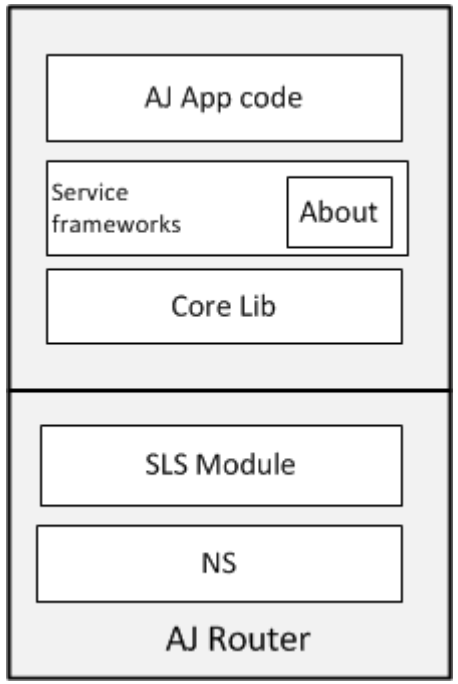
Name Based Discovery

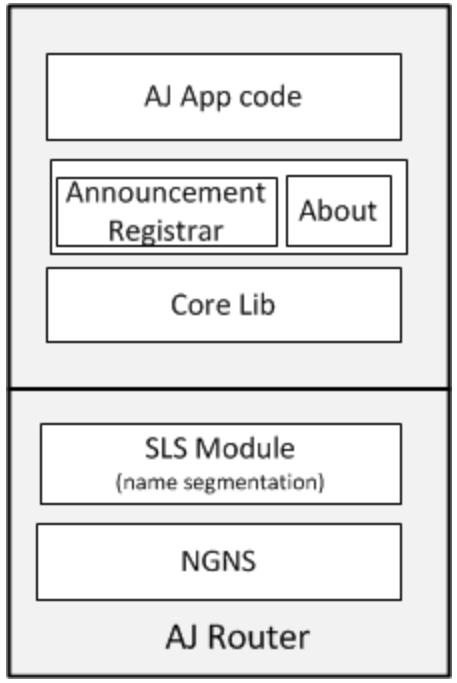
- **Test iteration**
  - Start 5 Advertisers, each advertising one name for a duration of 80 seconds, on five of the six Android devices *(all devices advertise with a common prefix)*
  - Wait for 45 seconds, to wait out the unsolicited advertisement traffic.
  - Start 1 Finder, each attempting to discover five names based on the common prefix for a duration of 37 seconds (essentially 80s - 45s, plus a safety buffer of two seconds)
  - Wait for the program to exit on all devices. The Finder reports each name that was found and how long it took to discover it.

Key metrics	**14.02 Estimate (*p* =0.35)**	**14.06 Estimate (*p* =0.35)**	**14.06 Android (sleep)**	**14.06 Android (no sleep)**
Prob. that discovery succeeds within 1 sec	12.3%	60%	41.9%	74.3%
Prob. that discovery succeeds within 10 sec	32.4%	97.4%	89.4%	99.9%

Protocol Stacks

- 14.02 vs 14.06

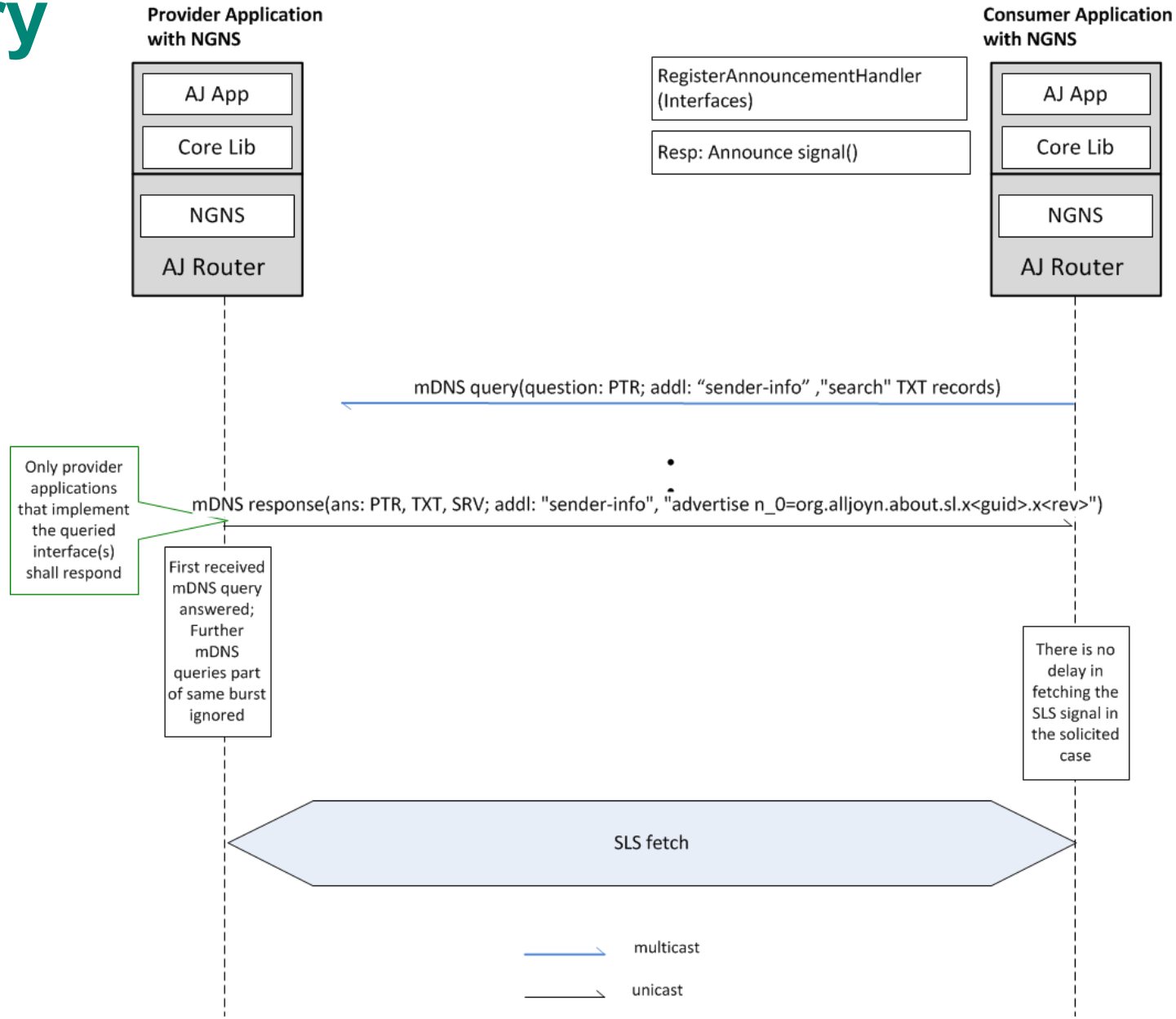




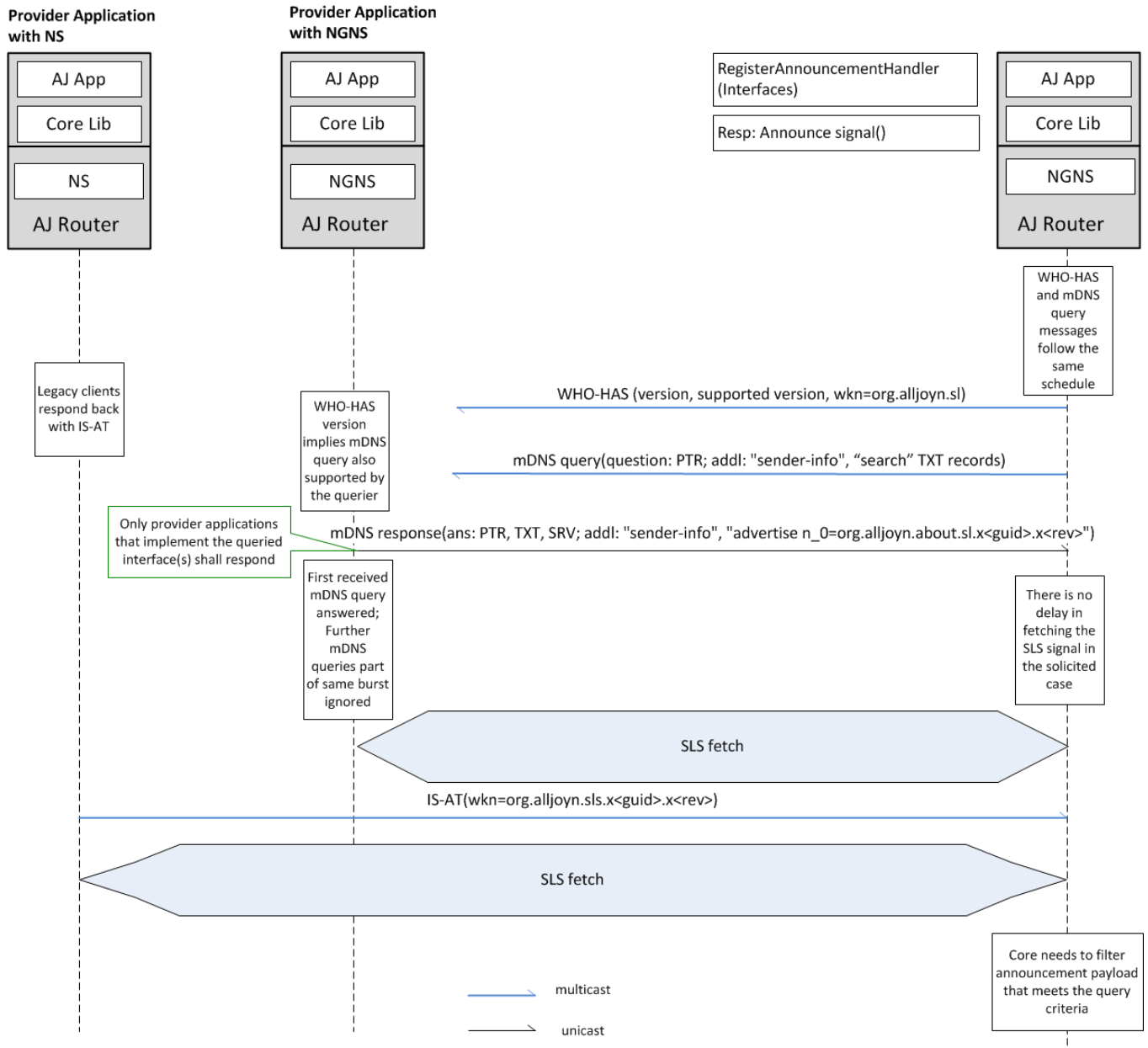
Discovery Functions

Discovery	Discovery use case	Pre-condition	API	Input	Output
Existing	Advertise WKN		AdvertiseName	WKN	
Existing	WKN based search	WKN is advertised	FindAdvertisedName	WKN prefix	WKN
Existing	Presence	FindAdvertisedName	LostAdvertisedName		WKN
Existing	Send Announce signal	Application provides Announce() payload	SendAnnounce()	Announce payload	Consumer App receives all Announce signals due to addMatch rules
New	Interface-name based search	Announce is being emitted by the Provider Application	RegisterAnnounceHandler()	Interface Name(s)	Announce signals that match the search criteria
New	On-demand Presence	Name has been discovered before	Ping()	name	Enumeration

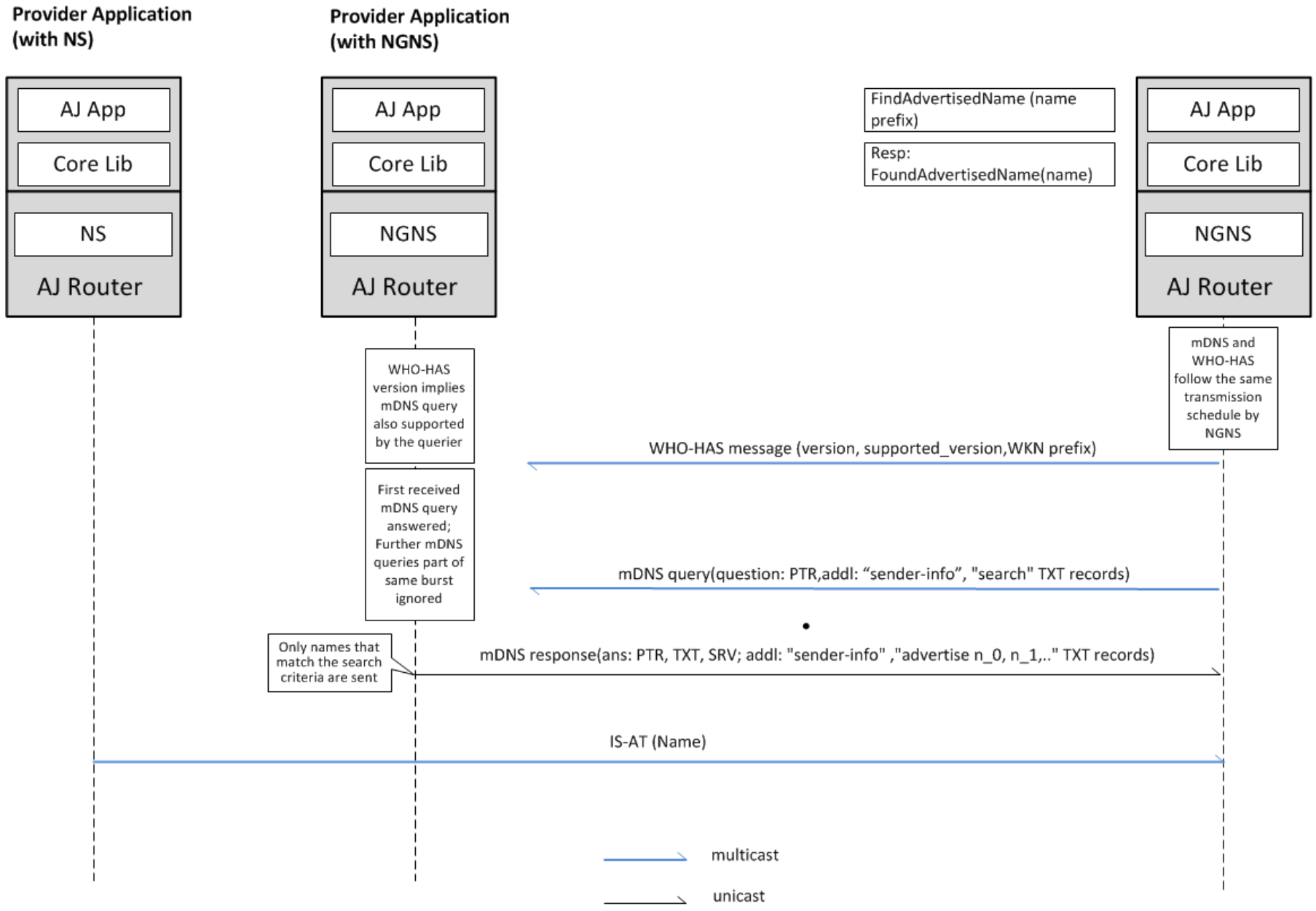
Interface-based query (NGNS-NGNS)



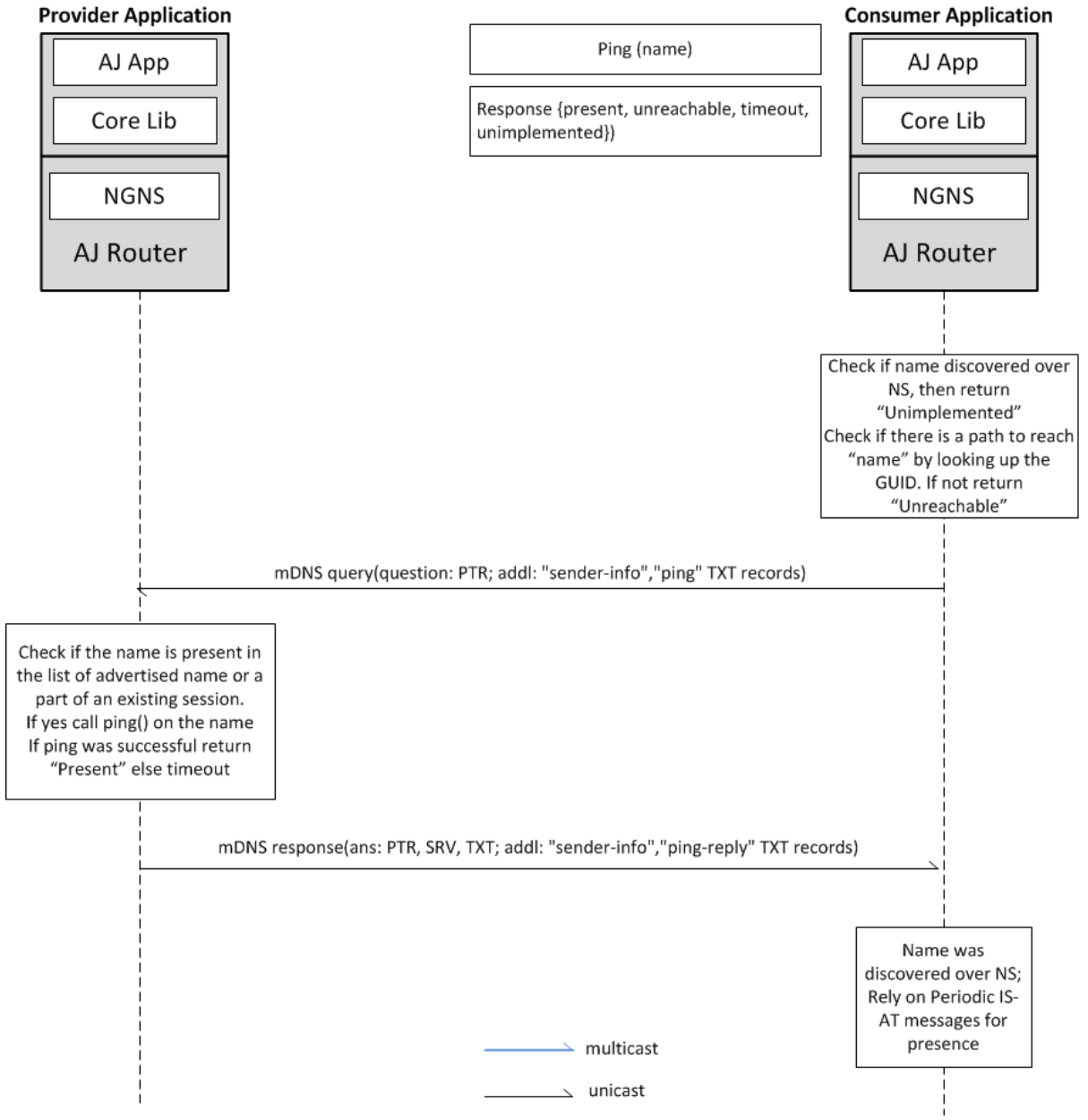
Interface-based query (NGNS-mixed mode)



Name Based Query (NGNS-mixed mode)



Presence – Ping API



Interface search: mDNS Query

- mDNS Query
  - sent to multicast IP and port as specified in RFC 6762 always and Subnet directed broadcast (if configured to true)
  - Top bit in the qclass field in the DNS header is set to indicate unicast responses are accepted
  - Question Section: _alljoyn._udp.local. PTR record
  - Additional
    - Name="search.<guid>.local." TXT [i_0] or [i_0,i_1,..] //if multiple interfaces are being queried; could be wild char *
      - Keys are ANDed
      - One search record per query
    - Name="sender-info.<guid>.local." TXT [pv, bid, guid, UDPv4, IPv4] //connection specs for the unicast NS
    - Where key nomenclature is as follows:
      - pv: version of the NS
      - bid: burst identifier (1..n where n is determined by the schedule) ← As per the schedule it is 1..5 in 14.06 release
      - guid: daemon guid
      - UDPv4: UDP port to be used with IPv4
      - IPv4: unicast IPv4

Examples Search Record

- Name="search.<guid>.local." TXT record
  - One Search record per query
  - Example1: keys are ANDed
    - txtvers=0
    - i_0=org.alljoyn.Clock
    - i_1=org.alljoyn.Radio

mDNS Response

- mDNS Response (unicast)
  - Answer
    - _alljoyn._udp.local. PTR record to domain name <guid>._alljoyn._udp.local.
    - <guid>._alljoyn._udp.local. TXT record [UDPv6] ← reflects UDPv6 port for AJ (D-D) over IPv6
    - <guid>._alljoyn._udp.local. SRV record UDPv4, <guid>.local. ← reflects UDPv4 for AJ (D-D) over IPv4
  - Additional
    - Name="sender-info.<guid>.local." TXT [pv, bid, guid, UDPv4] ← reflects where unicast polling over UDP is sent using NS; IP addresses are sent in the A and AAAA record
    - Name="advertise.<guid>.local." TXT [n_0=org.alljoyn.about.sl.x<guid>.x<id>]
    - <guid>.local. A record for IPv4 ← reflects IPv4 for AJ session (D-D) over IPv4
    - <guid>.local. AAAA record for IPv6 ← reflects IPv6 for AJ TCP session (D-D) over IPv6
    - Where key nomenclature is as on the previous slide
    - Address records always populated in unicast responses

NGNS in nutshell

- NGNS Discovery protocol is based on DNS-SD over mDNS
- Completely backwards compatible with 14.02 APIs and wire protocol
- Interface based query and Presence APIs are two key additions to the API set
- Subject to much stricter latency objectives

- https://allseenalliance.org/docs/api/cpp/classajn_1_1_bus_attachment.html
- https://allseenalliance.org/docs/api/cpp/classajn_1_1services_1_1_announcement_registrar.html

Technical Steering Meeting

Thank you

Follow us on