Experiments in Cognitive Radio and Dynamic Spectrum Access using An Ontology-Rule Hybrid Architecture

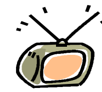
Allen Ginsberg, Jeffrey D. Poston, and William D. Horne

The MITRE Corporation

McLean, Virginia

USA

Background

- **What is a radio?**
  - **Anything that utilizes spectrum:** , ,  **(transmit-receive)**
  - **State-of-the-art: *Software Defined Radio* (SDR)**
    - **uses software for the modulation and demodulation of waveforms**
    - **SCA reference model (CORBA & POSIX based)**
- **Spectrum *Allocation***
  - **Spectrum use regulated by each nation & international treaty**
    - **key governing principle: avoid interference.**
    - **use static allocation scheme**
      - **e.g., Analog TV uses UHF, VHF bands, etc.**
- **Has spectrum become scarce?**
  - **More an issue of inefficiency of static allocation**
    - **there is plenty of unutilized spectrum at any given place and time**
- **Dynamic Spectrum *Access***
  - **Could be compatible with allocation to primary (licensed) users**
  - **Allow secondary users to access when primary is not using**

MITRE

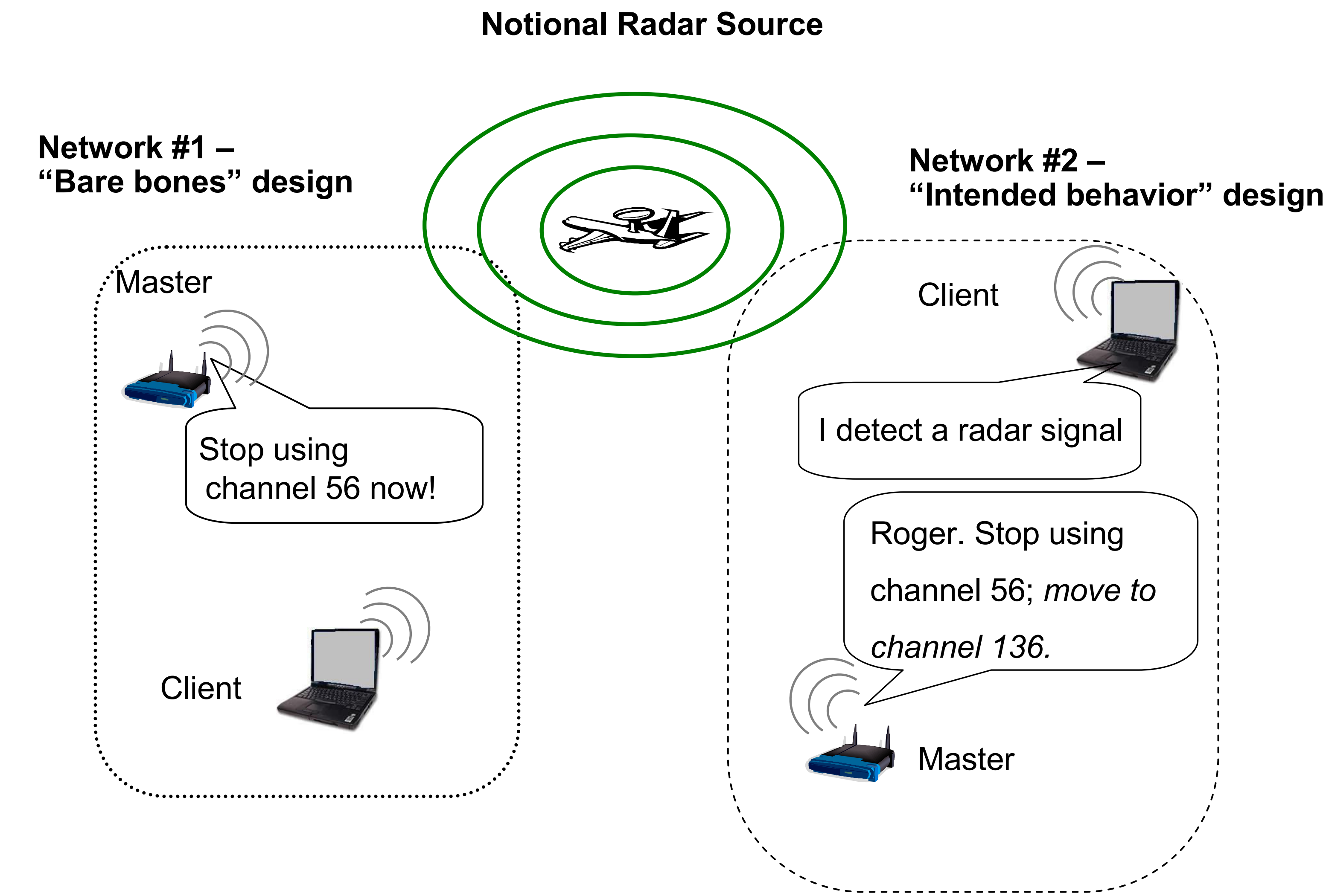
What is *Cognitive Radio*?

- *At least the following:*
  - Device architecture & infrastructure for dynamic spectrum access
  - Policies & conventions that allow CRs to coexist
- Many research approaches, e.g., game theory
- Our approach: use semantic technologies
- Why?
  - Build upon DARPA XG approach to policy checking.
  - Spectrum policy will evolve, not disappear.
  - Dynamic access must coexist with those policies.
    - All radios must be policy-compliant (behavior *guided by* policy)
    - Formal representation of policy allows flexibility, etc. in CR development
  - Radio that understands policy can do more
    - *Knowledge-driven differential-response*

Case Study: Dynamic Frequency Selection (DFS)

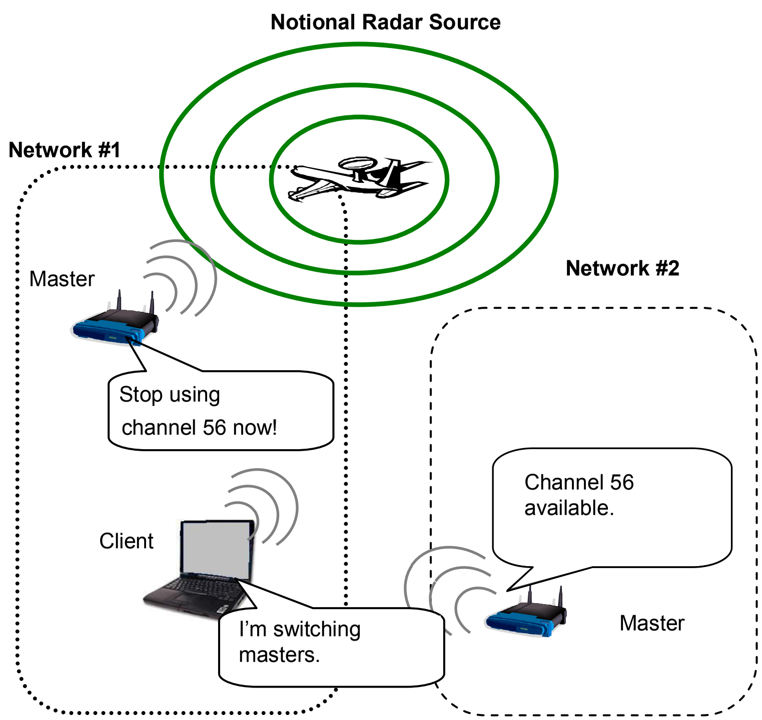
- Allows 5GHz wireless devices to coexist with legacy radar
- The FCC DFS rules require 2 device types: Master & Client
  - Client
    - must *associate* with a Master before using channel in band.
    - can only use channel if Master says “ok.”
    - must vacate channel if Master sends vacate signal
    - *if* client has radar-sensing capability, it must report detected radar to associated master (and then vacate channel)
  - Master
    - must listen for 60 sec for radar prior to allowing first use of channel and refrain from permitting use if radar detected
    - monitor for radar for channels in use
    - if radar detected, broadcast vacate signal to clients.
    - Channel must be vacated within 10 seconds of radar detection
    - refrain from using the channel after radar detected for at least 30 min.

Basic DFS Design Profiles - Scenarios

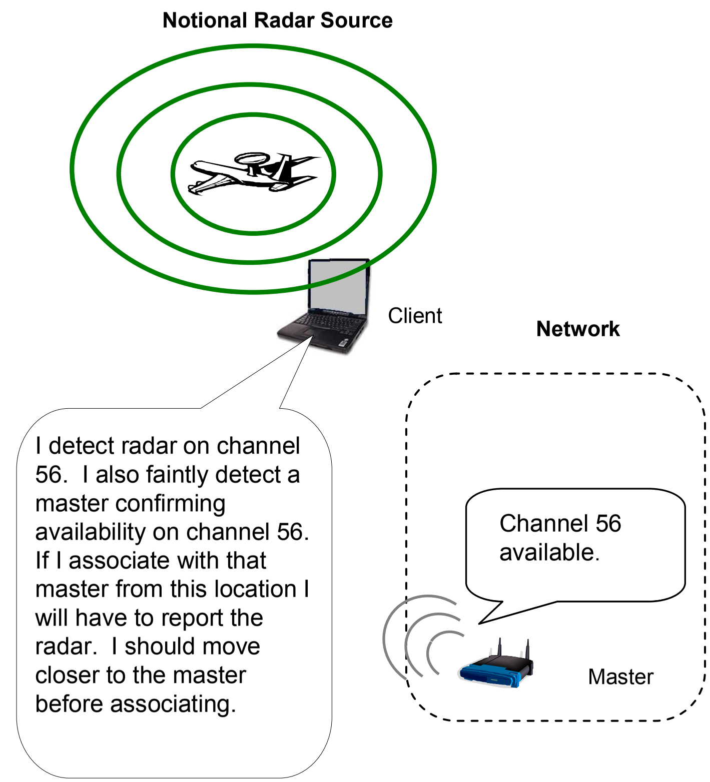


Enhanced DFS Design Profiles - Scenarios

(Involving Knowledge-Driven Differential Response)



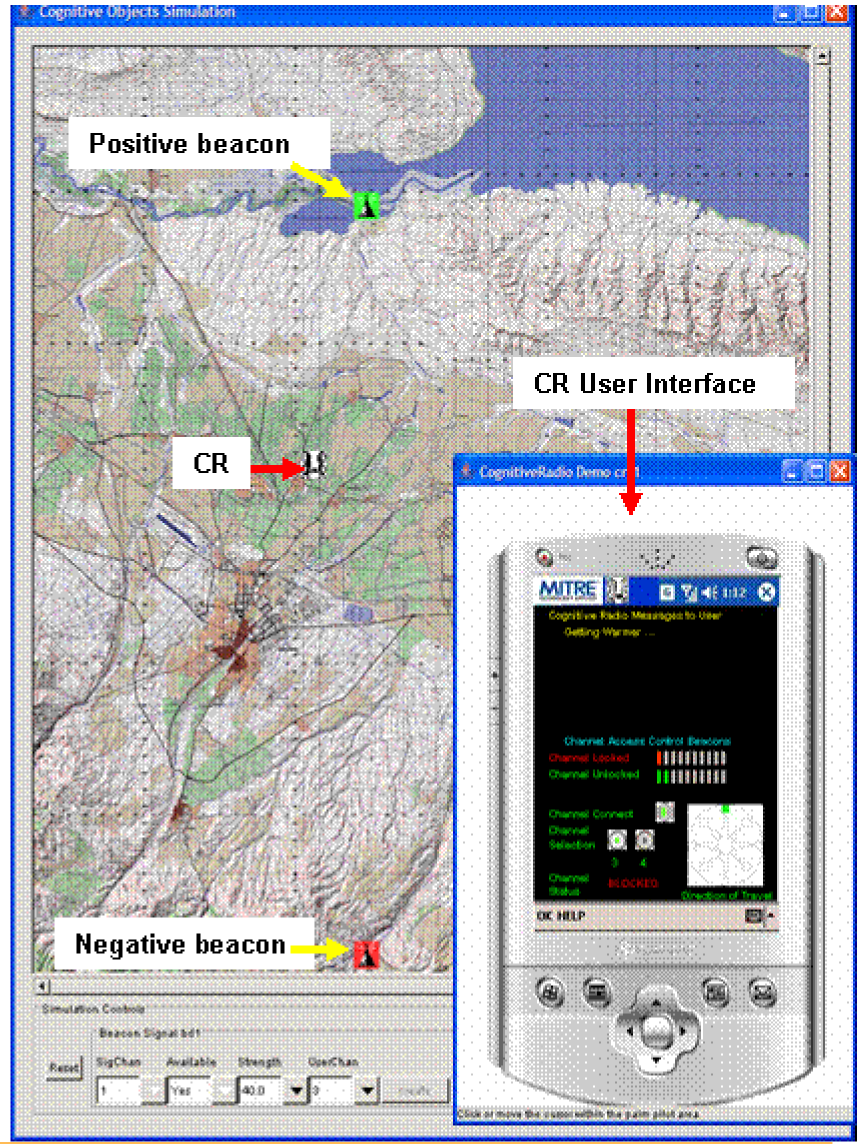
Enhanced DFS Scenario-1: Master-Swapping



Enhanced DFS Scenario-2: Gradient-Surfing

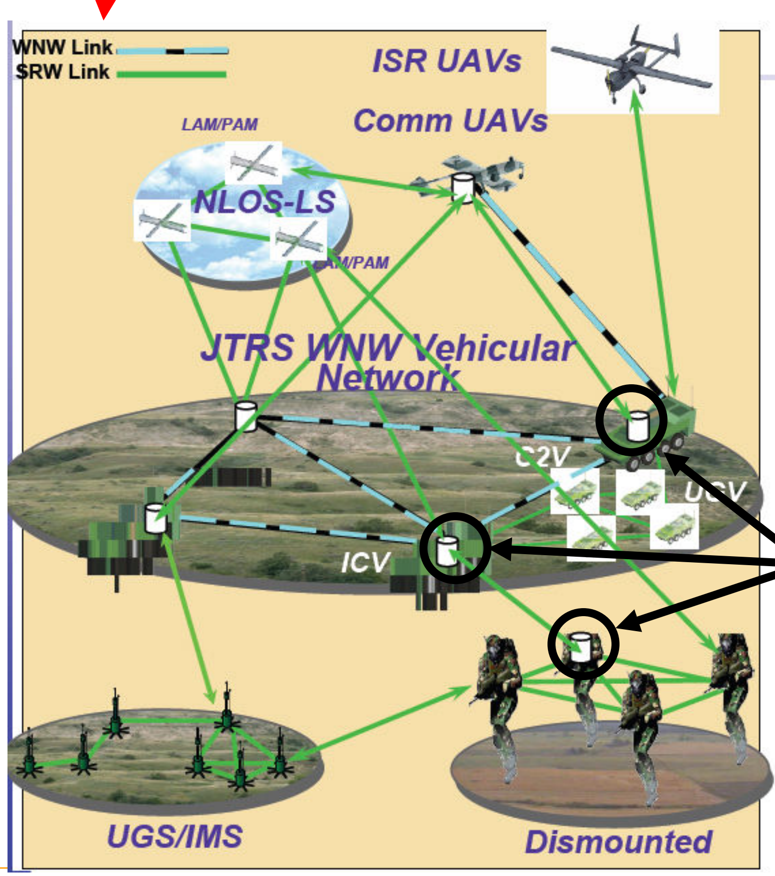
Prototype Simulation using Beacon-Based Spectrum Access

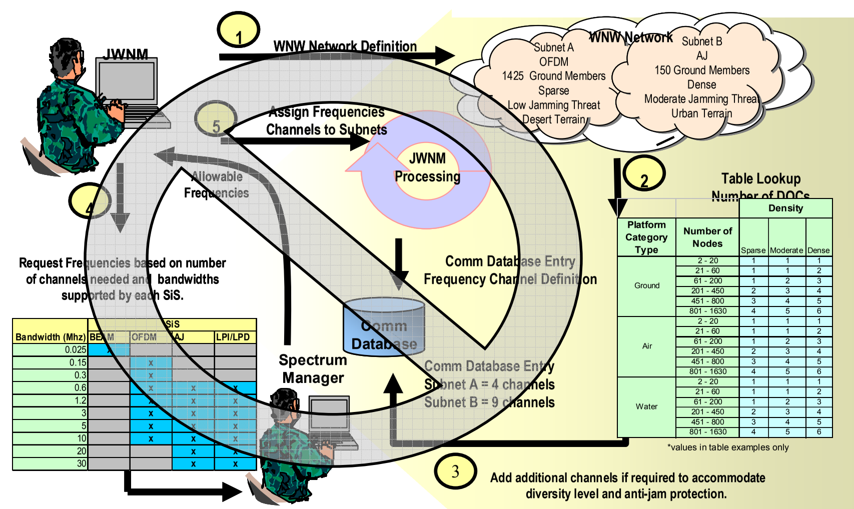
- OWL Ontology
- Jena rules
  - use of custom built-in feature
- Radio hardware and radio propagation emulated in Java;
- In terms of DFS scenarios
  - positive beacon is Master signaling channel available
  - negative beacon is radar signal (or master sending vacate signal)
- "Semantic beacons" could be important in future CR apps



Example Tactical Military Scenario

- **WNW and SRW (and other tactical waveforms) need significant spectrum → Competing for same available spectrum!**
  - **Current spectrum management processes not sufficient for dynamic, multi-waveform environment**





- **Embed "cognitive" modules into radios to provide:**
  - **Automated frequency selection & deconfliction**
  - **Adaptive to changes in mission needs and environment**

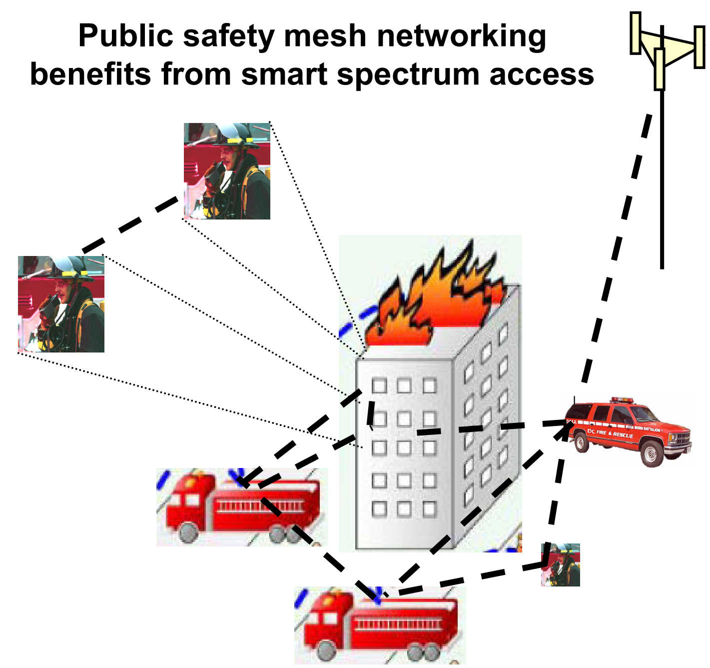
Smart Radio Could Improve Response During Complex Public Safety Events

- **First responder effectiveness requires improved communications capabilities & interoperability**

Cognitive Radios can help address the following issues*:

- Multiagency spectrum assignments & coordination
- Network control and management
- Predefined user configurations
- Incident-specific configurations including "scene-sense" automatic configuration
- Embedding policy considerations at the communications device level

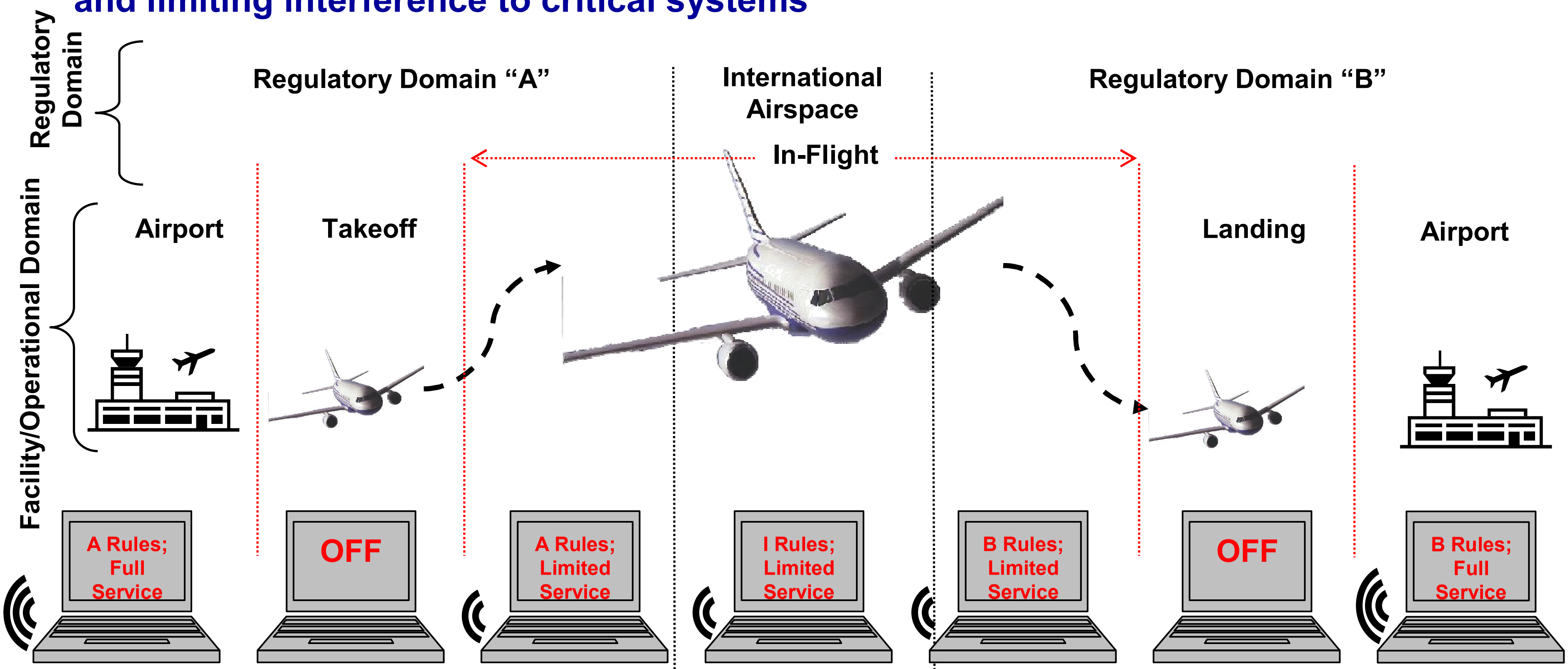
Public safety mesh networking benefits from smart spectrum access



* List derived from presentation by John S. Powell, Chair, National Public Safety Telecommunications Council (NPSTC) SDR Working Group, at Cognitive Radios Conference, Las Vegas, NV, Mar 2004

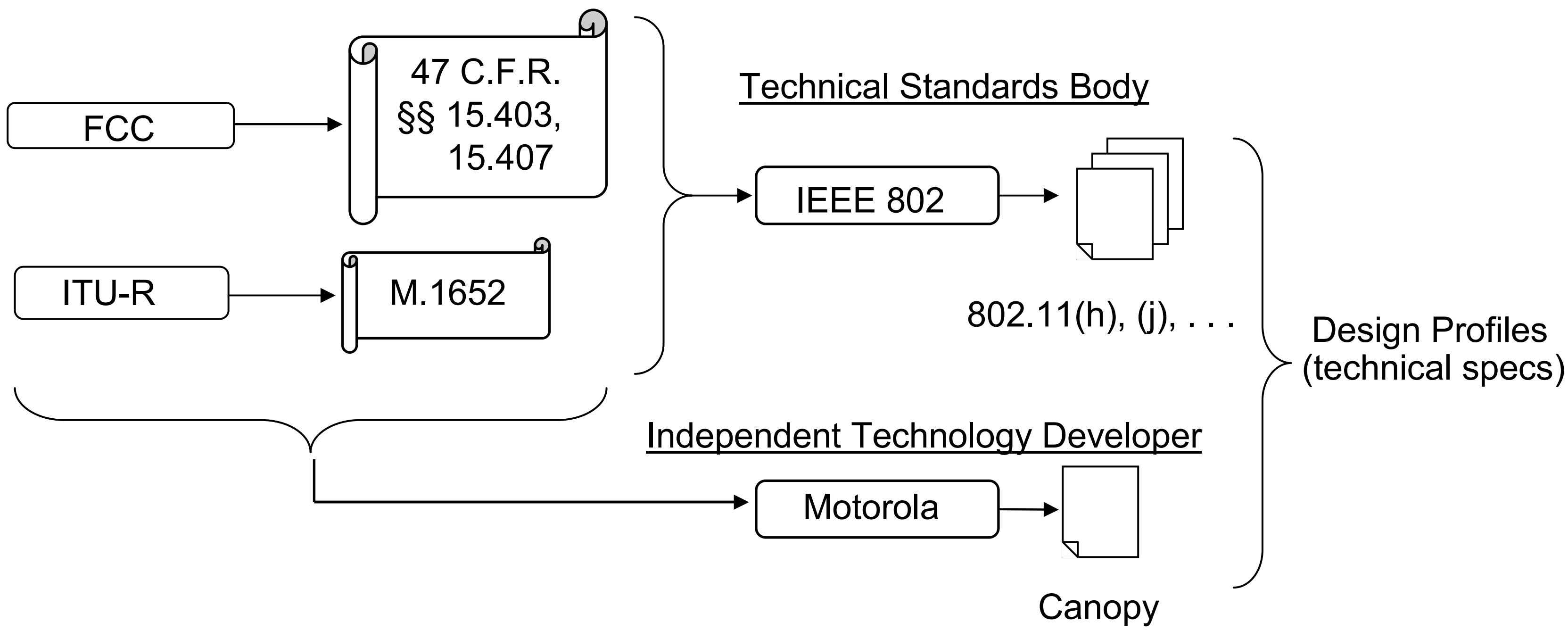
Context-Aware Radios Could Improve Both Safety & Customer Experience

- Domain aware radios, aviation and passenger systems (Portable Electronic Devices (PEDs)), can adjust behavior during a flight ensuring compliance with regulations and limiting interference to critical systems



Portable Electronic Device (PED) (future versions of current systems under review in RTCA SC-202) understands "context" and operates according to domain—includes mandatory override/shutdown to ensure protection of avionics

Present Day Community Approach to DFS Interpretation & Implementation

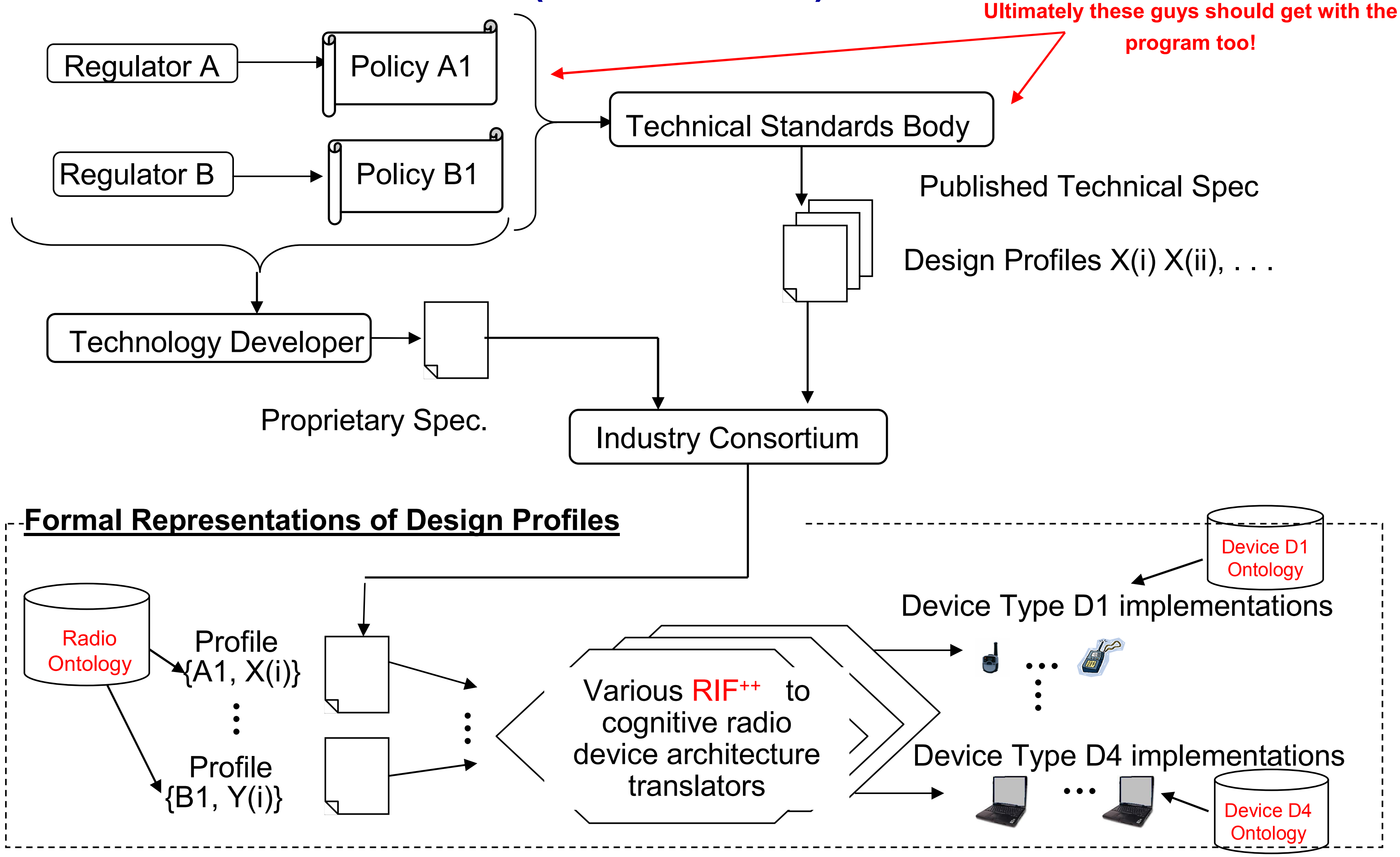


DFS policy/technical-specs required 4+ years to develop;

This pace of rulemaking is a barrier to rapid innovation in the CR arena.

Cognitive Radio Community Architecture

(RIF Use Case)



Formal Representation of A DFS Design Profile

- Suppose after sensing radar all activity can be terminated within 1 sec…
- Then checking for radar every 9 sec suffices

```
R1: If
      ChannelInUse(?ch,t_now)
      LastRadarCheckTime(?ch,?t)
      t_now - ?t >= 9
    Then
      Required(radar_check,?ch, t_now)

R2: If
      RadarDectected(?ch,?t)
      t_now >= ?t
      t_now - ?t <= 9
      ChannelInUse(?ch, t_now)
    Then
      Required(vacate_signal,?ch, t_now)
```

– These are deduction rules: *statements* about what this design requires. They do not *cause* a device to perform the required actions.

– *radar_check, vacate_signal* are constants. They denote "abstract" actions

– t_{now} is a constant denoting the "current instant of time"

– ?ch, ?t are variables over channels and instants of time.

The semantics of the above (and the predicates) are either defined in an ontology and/or by other axioms.

Ontology+rules form precise representation of *what* is required, doesn't say *how* to implement.
e.g., use DB for keeping track of radar check times? How far back in time?

Observations

- No “impedance mismatch”, in principle, between rules and ontologies at this level
- Rules are simply “axioms” built upon the deductive backbone of well-defined concepts in the ontology
- Standard model-theoretic semantics determines what follows from what
- No instance management or epistemological issues *per se*
  - no need to consider non-monotonicity, e.g., CWA
  - time/temporal-change viewed *sub specie aeternitas*
  - …

because this is a *specification* of what “laws” will hold true if everything works as intended by this design. The whole point is to capture those intentions.

Implementation Level: Rules to *Guide* Device Behavior

```
ECA-1:
 ON
    // this alarm rings every 9 secs
    RadarCheckAlarm.isRinging()

 IF ChannelInUse(?ch)

 DO
   {
      result = CheckForRadar(?ch);
      if
         (result == RADAR_FOUND)
      then
         BroadcastVacateSignal(?ch);
   }
```

– An imperative (ECA) rule: invokes actions, *causes* things to happen.
– Ideally, would like RIF++ based translator to somehow "translate" R1, R2 into ECA-1.*

Question: what happened to *time*?

General Answer: different ontology at this level. For example, no such thing as a *RadarCheckAlarm* at design level.

Also *BroadcastVacateSignal* **invokes a procedure in the SDR implementation: cannot expect complete semantics for it.**

*Our RIF use case does *not* require this notion of translation: translation *within dialects* will also benefit collaborative development of cognitive radio.

Observations

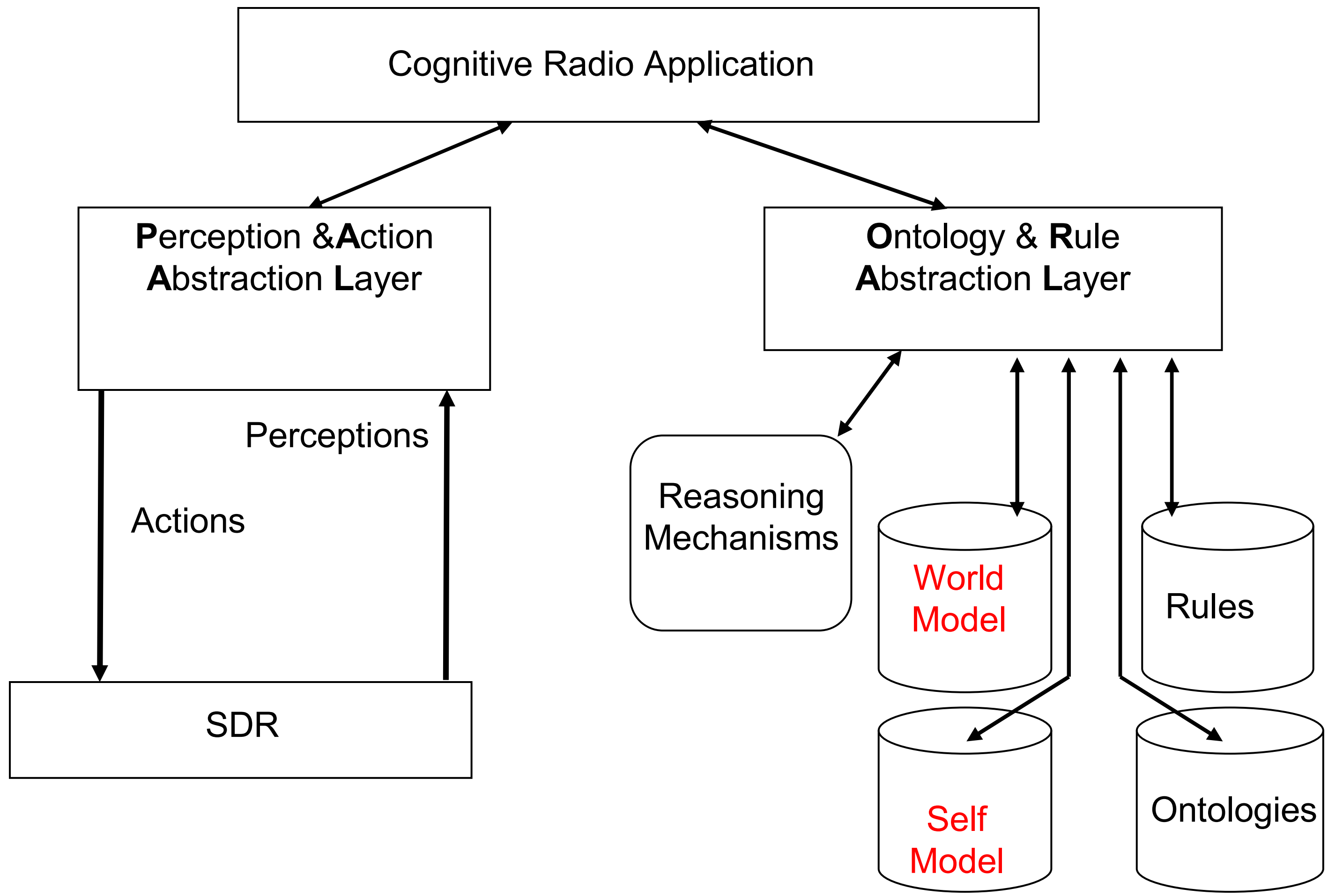
- Implementation ontologies different from design level
  - Reduction of higher level theoretical notions to observationally/operationally defined/measurable notions.
  - e.g., at higher level notion of *radar* involves some reference to function or purpose; at device level concern is how to detect certain features of waveforms.
  - How much higher level stuff to put in device level? (tradeoffs)
- Rules at device level tend to be imperative, e.g., ECA
- The "arrow of time" is real at this level
- Standard model-theoretic semantics is not the whole story
- Instance management issues
- Epistemological issues

Key issue in a nutshell: how (and how much) to preserve relationships

among policy & design worldviews and implementation worldview?

Cognitive Radio Architecture

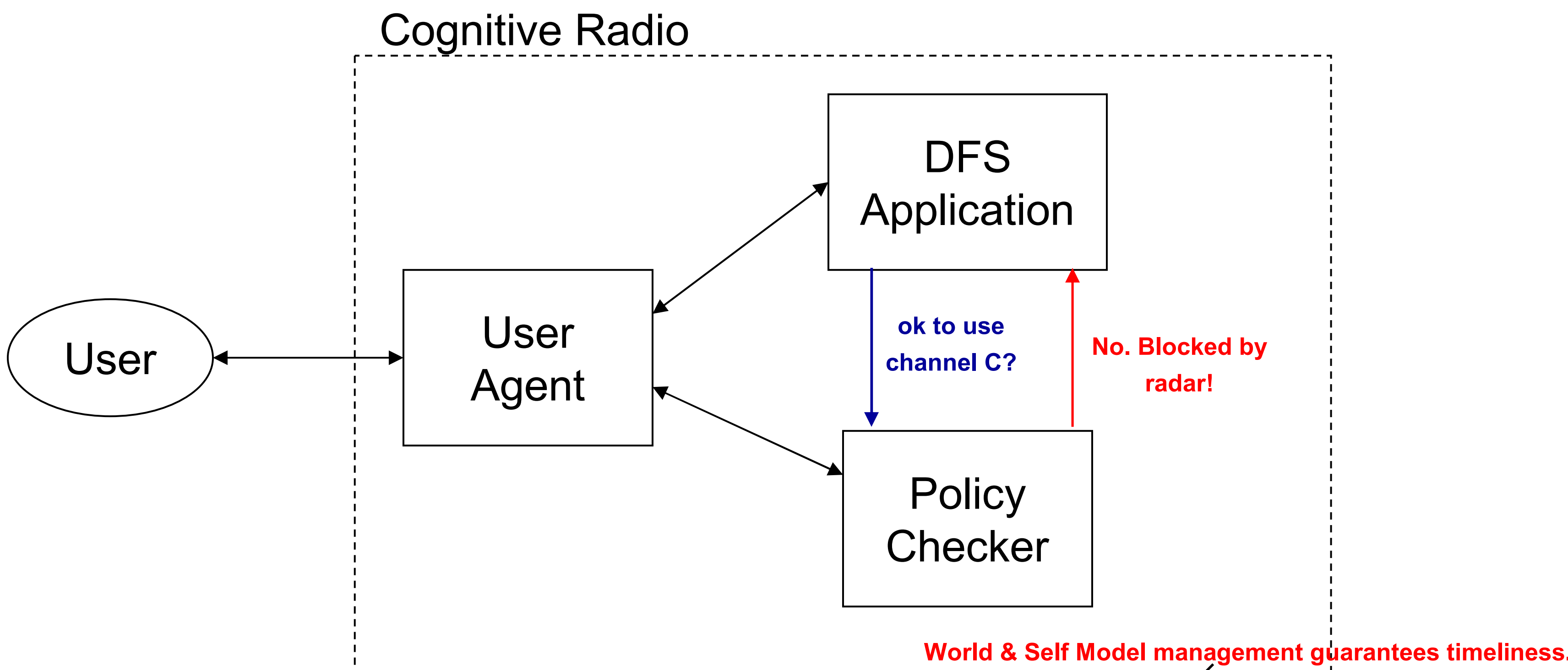
(Transcends design and implementation levels)



Device Architecture: World Model & Self Model

- World Model is the “external world” according to the CR
  - what things are present, facts about its user, etc.
- Self Model is what the CR knows about itself
  - current goals, capabilities (e.g., radar sensing), etc.
- Management of world & self models
  - what goes in, how it gets it, how long it stays, and how it gets removed
  - obviously must be sensible with respect to the rules, ontology, and reasoning mechanisms that CR uses
  - this will be determined to a great extent by the underlying SDR and the PAAL layer

Device Architecture for Enhanced DFS Service



received when? any contrary info received after?

If

the goal of using channel C is blocked by the presence of radar-signal r with heading h1
and s is a control-signal with heading h2 received from Master m implying channel C availability
and h2 and h1 are sufficiently distinct

Then

Attempt to move in the direction indicated by h2 until the signal strength of r is below the required threshold
and then attempt to associate with m

Thoughts…

- **When thinking about ontology+rule interaction**
  - **Different levels entail different considerations**
    - **Community architecture (policy) & design profile levels**
      - **Think in *sub specie aeternitas* mode: rules+ontology together describe what *is*.**
      - **How to make it possible to reason *about* policy & design profiles**
    - **Implementation level**
      - **Designing something whose behavior is guided (in a causal sense) by policy**
      - **Guarantees need for rules with imperative semantics**
      - **Be clear about the structure of self and world models and how managed: this will help to figure out how rules should be written.**
- **Relations among ontologies at different levels fundamental to “real” understanding**
  - **not necessarily needed in app.**
  - **but essential to RIF^{++} !!!**

MITRE

Closing Remarks

- The regulatory community, tech standards orgs, and the technology developers must have some shared understanding of how to formally specify CR behaviors;
- A Cognitive Radio Community Architecture based on semantic technologies offers a framework for developing this shared understanding;
- This requires knowedge representation in the form of both ontologies and rules;
- RIF offers the means to decouple the community’s understanding of the rules from specific rule-based inference engines;