1. Highly-scalable, cloud-capable, shareable decision support service.
2. Extensible data adaptor architecture supports multiple data input formats, including interoperable C32 Continuity of Care Document (CCD).
3. Architecture agnostic to underlying decision architecture, may provide multiple back-end decision engines providing different computing.
4. Terminology classification and normalization services support multiple input terminologies transparently to users.
5. Expressive XML action model returned from service, allowing users to customize display of actionable recommendations - declarative output does not dictate final display to end-users.

Comments on DSS experience:

- Decision support service approach most pragmatic, expedient method to share executable knowledge.
- Success of DSS lies in hiding substantial complexity of decision support from EHR implementers.
- Legislating the availability of service insertion points to facilitate integration with a DSS will accelerate the adoption of shareable executable knowledge through remote execution.
- The use of standard summary of care documents as payload substantially reduces the barrier for using shared services.
- Early barriers to DSS include limited experience with CCD which should quickly abate with experience.
- Semantic interoperability major barrier to shareable CDS, and an area for continued investment.
- Availability of standard national CDS value sets critical to any knowledge sharing, and a strength of the PHS implementation.
- Definitional value sets and definitional rule specifications are highly reusable components for CDS.
- Successful reuse of CDS building blocks, necessary to achieve scale, rely on the existence of an underlying rules architecture, which underpins a framework such as ECRS.
- Representation of extensible, maintainable executable knowledge L4 may and will likely differ substantially from L3 specification.
- Problem solving methods applied to an L3 to create L4 are not well characterized and present a barrier to robust sharing at the executable level.
- HL7 datatypes, as used in the VMR, are suboptimal for use in an inference model, for example in representation of time or dimensioned entities such as observation values.
- Limited and inappropriate datatypes create unintended barriers for knowledge engineers in creating CDS.
- Current models such as the vmr, do not address the additional classes necessary to support problem solving methods beyond brute force combination of the available clinical classes. Intermediate classes quickly become necessary in all but the most trivial CDS.
- A successful vmr will ultimately be a set of class interfaces, and not simply datatypes, proven useful in supporting clinical inference over a representative set of CDS problems.
- The CDSC and opencds experiences demonstrate that sophisticated real world CDS is achievable through the use of contemporary business rules mgmt systems, with the promise of reuse if accompanied by an architecture for the rules themselves. The ruleML experience demonstrates that it is difficult to share executable representations among different implementations.