Formal Verification of Networks
—Description Language—

Peter Höfner

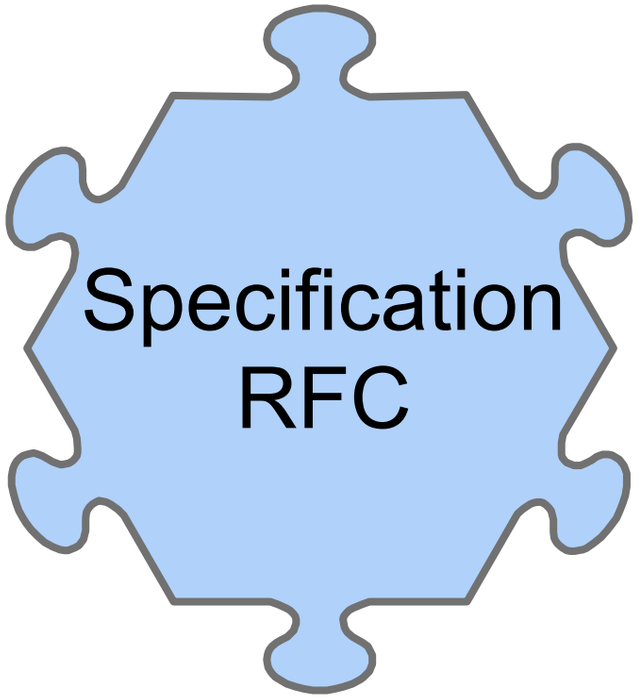
joint work with:
R. van Glabbeek, M. Portmann,
W.L. Tan, A. Fehnker, A. McIver

Australian Government
Department of Broadband, Communications and the Digital Economy
Australian Research Council

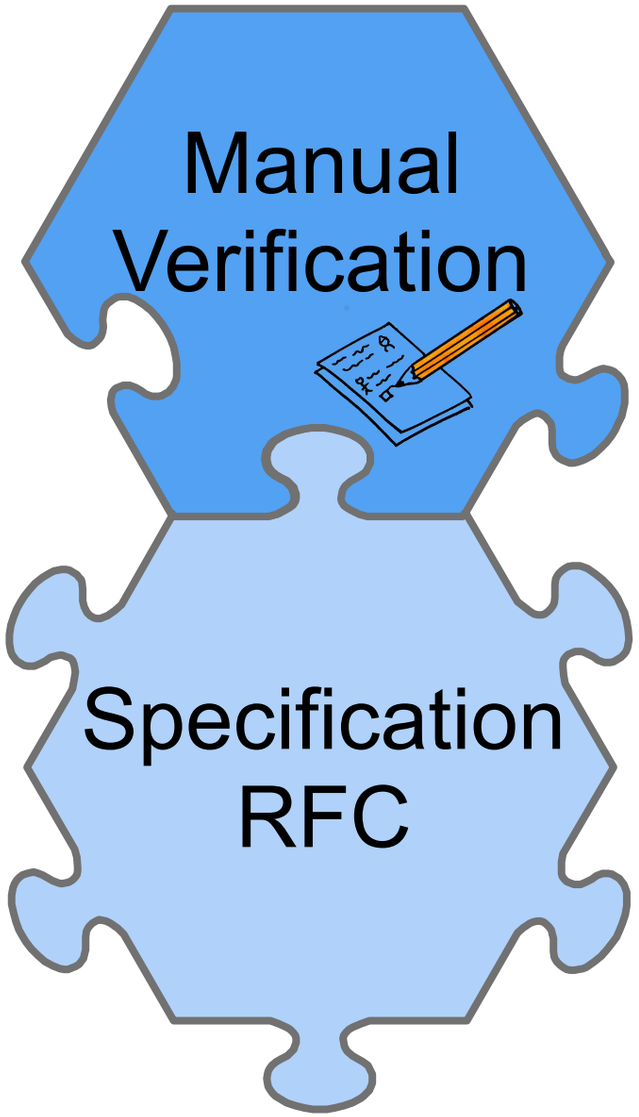
NICTA Funding and Supporting Members and Partners

The Big Picture

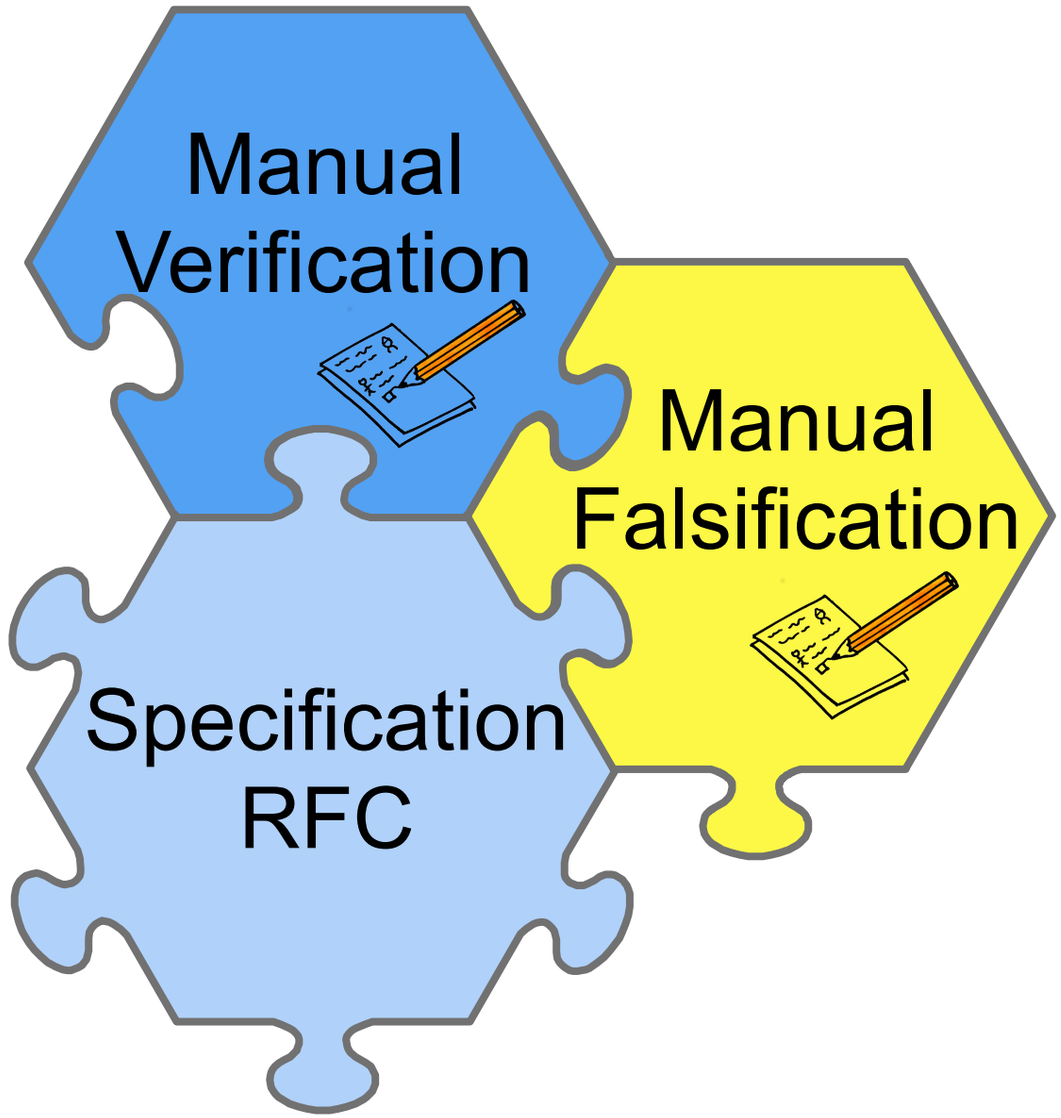
The Big Picture



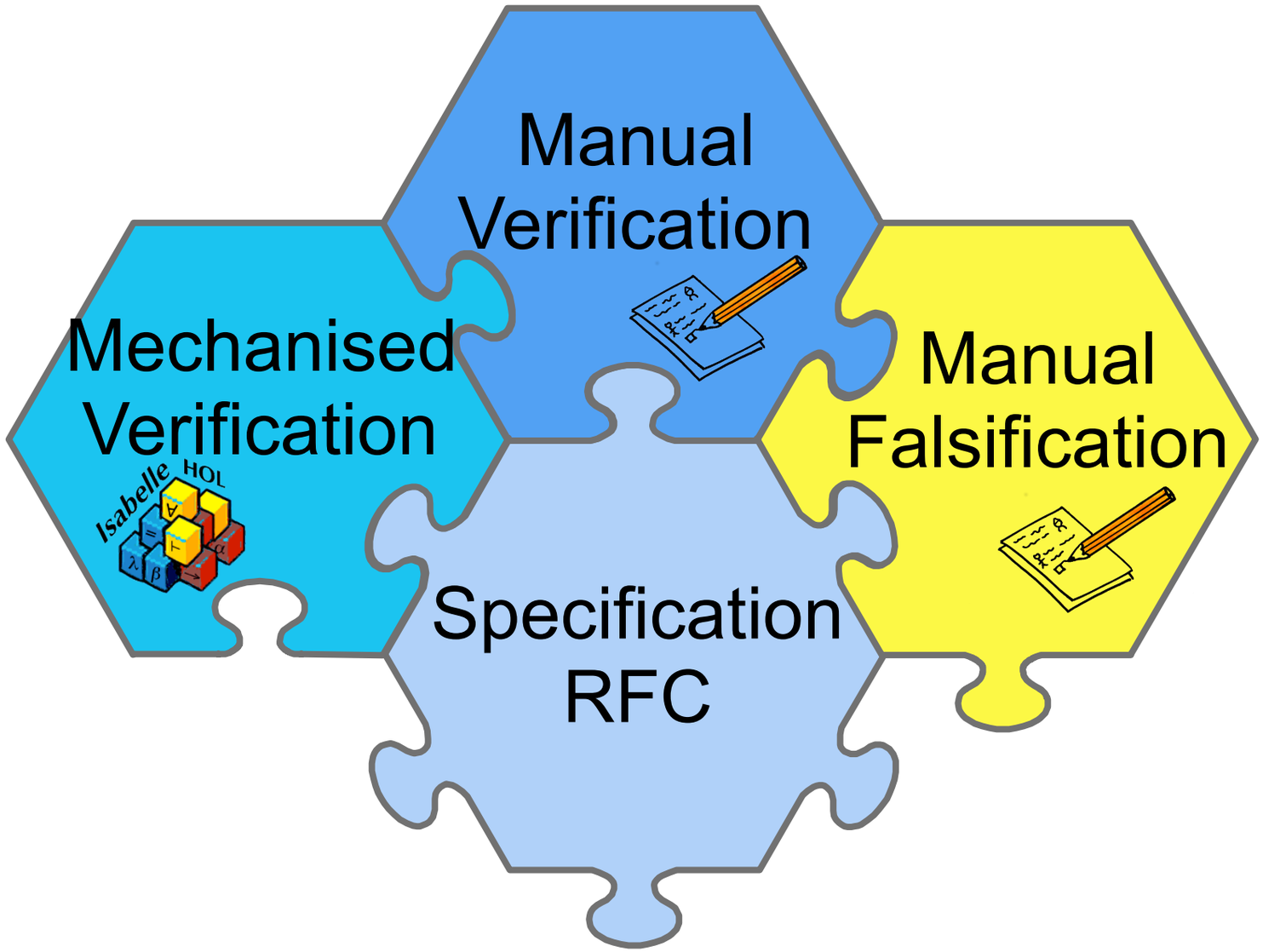
The Big Picture



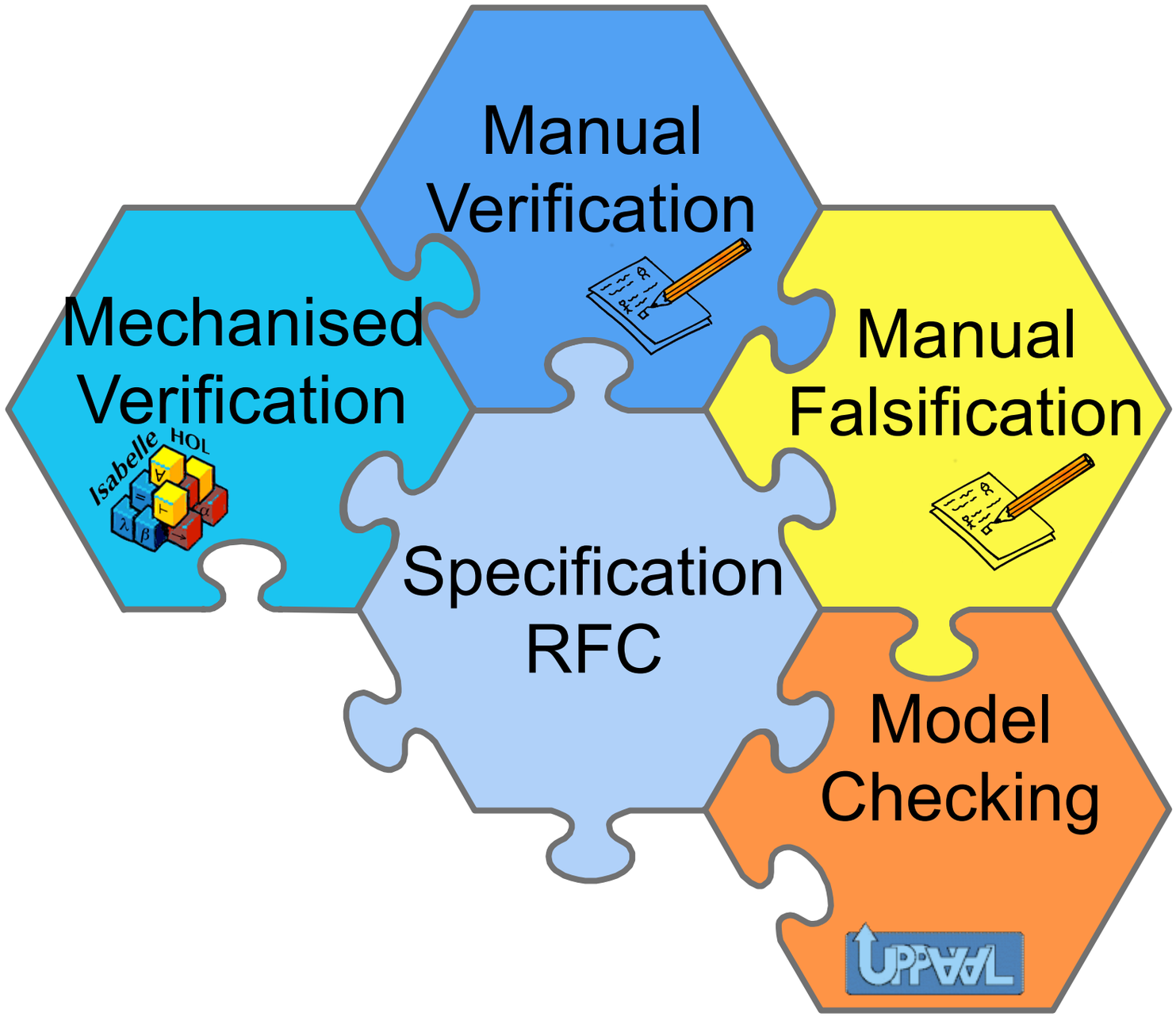
The Big Picture



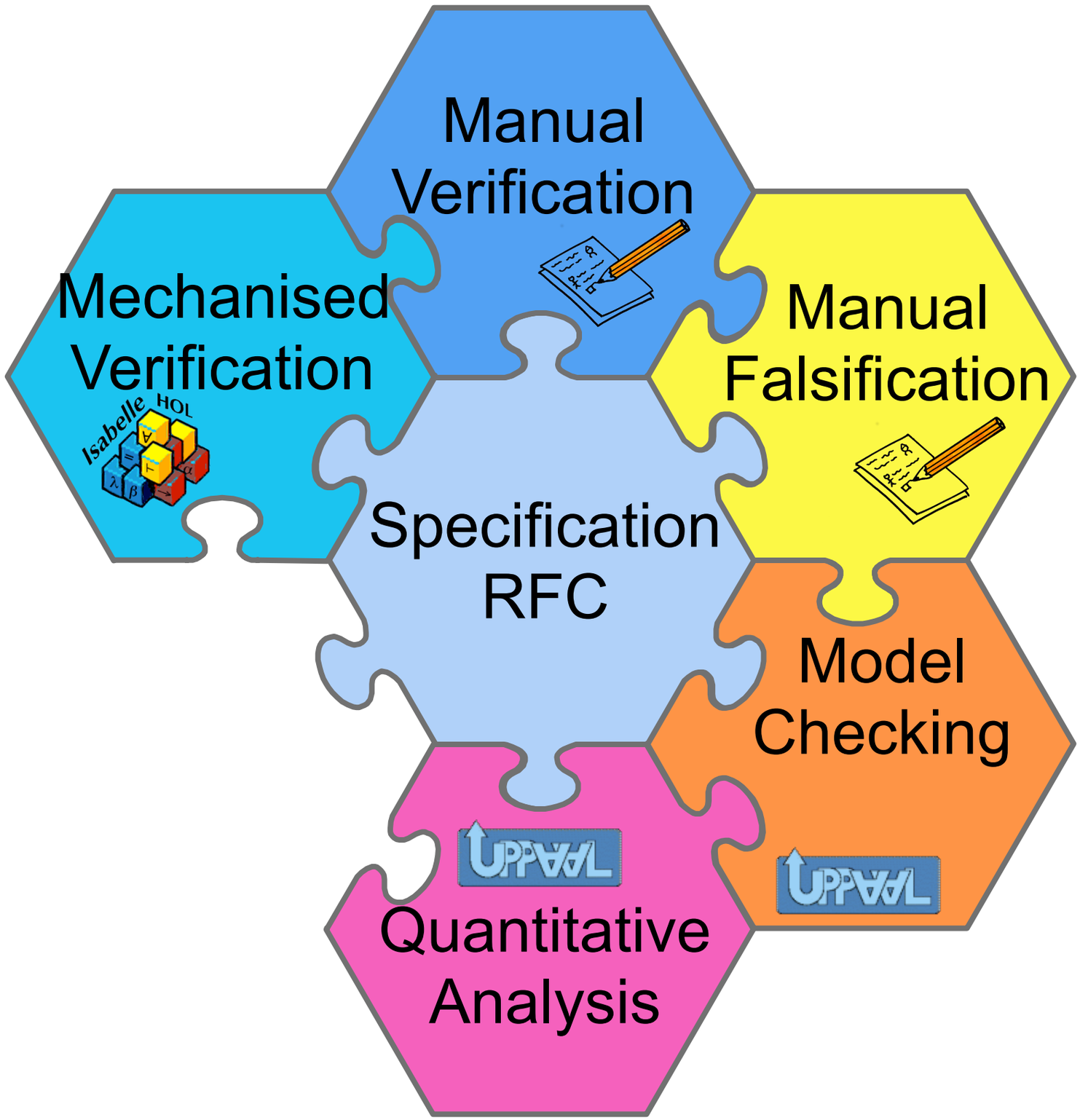
The Big Picture



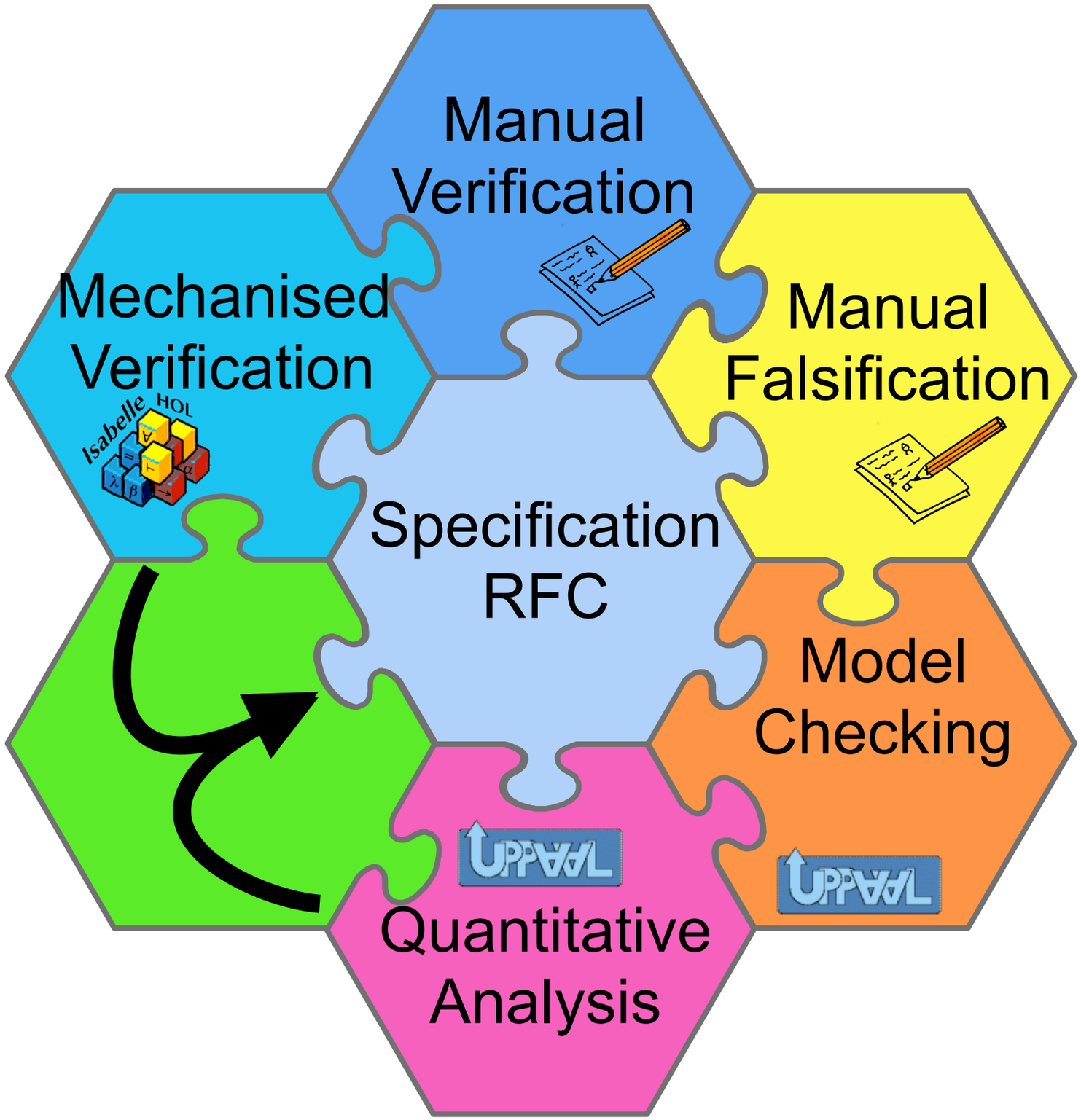
The Big Picture



The Big Picture



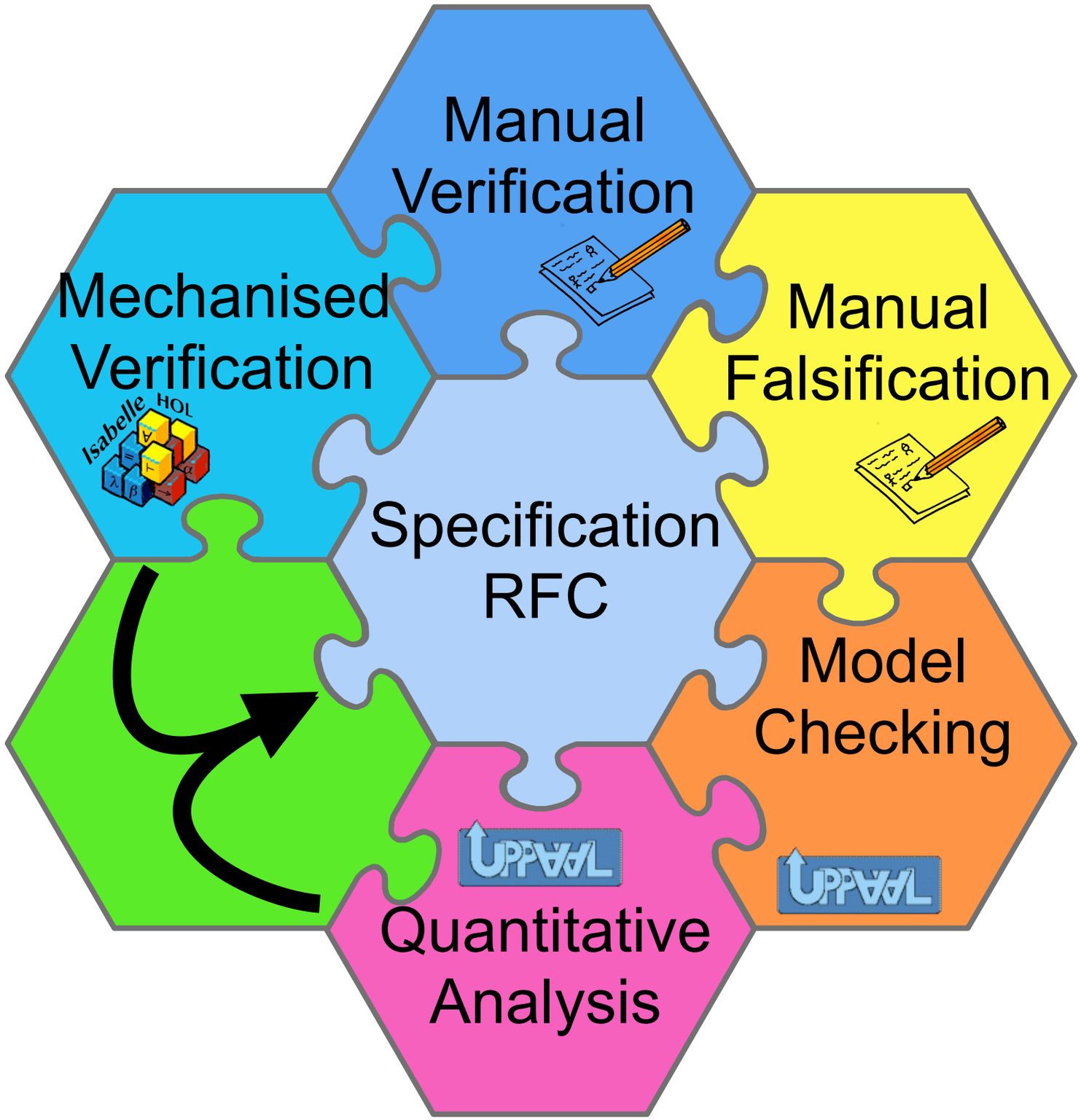
The Big Picture



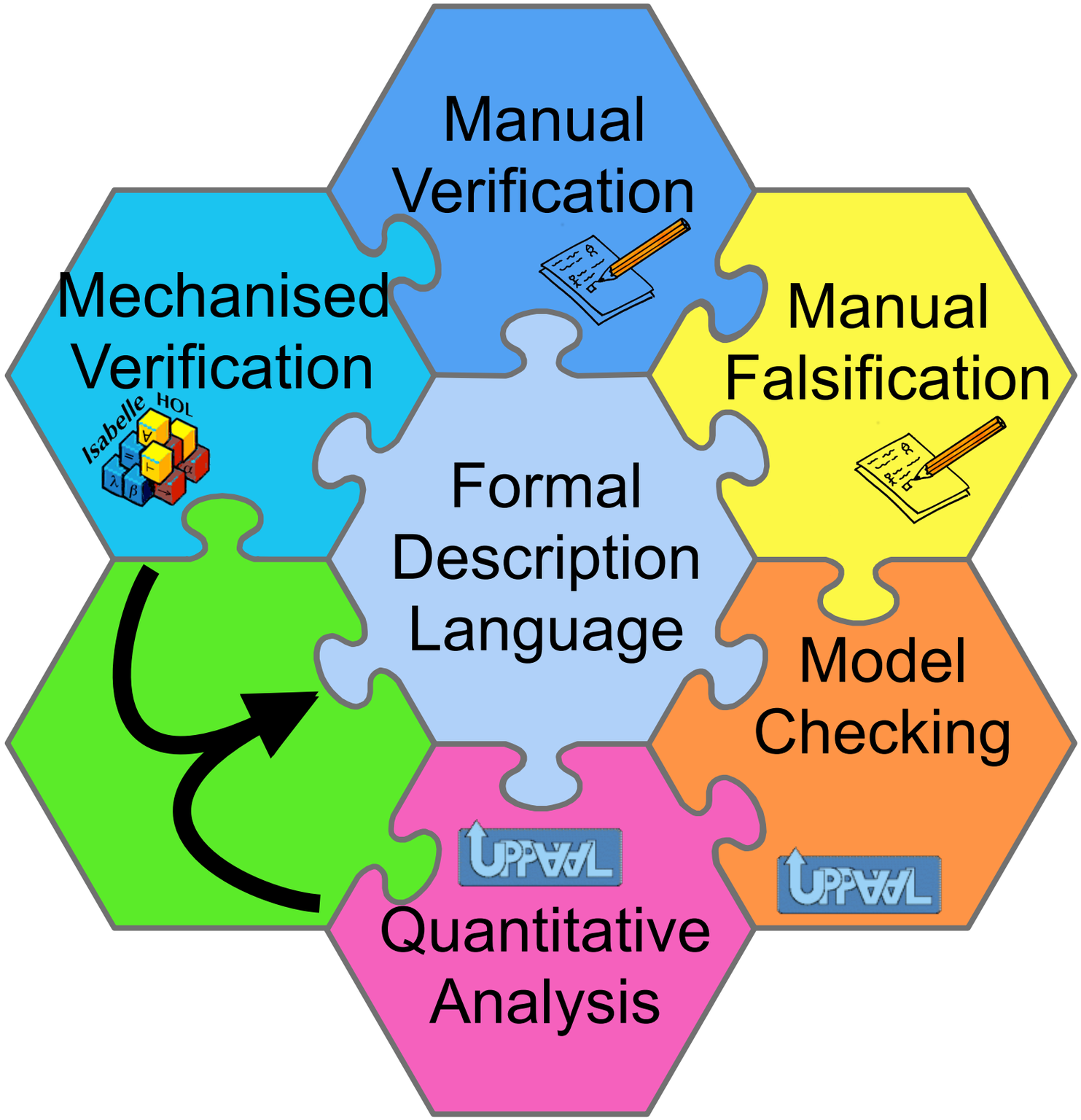
Motivation (Current Problems)

- specifications are not precise
  - written in English
  - ambiguous (sometimes incomplete)
- compliant implementations
  - have different behaviours
  - are not compatible
  - have serious flaws
- traditional evaluation techniques: simulation and test-bed
  - expensive
  - limited to (a small number of) specific scenarios
  - error found after years of evaluation
  - barely offer any guarantee for properties

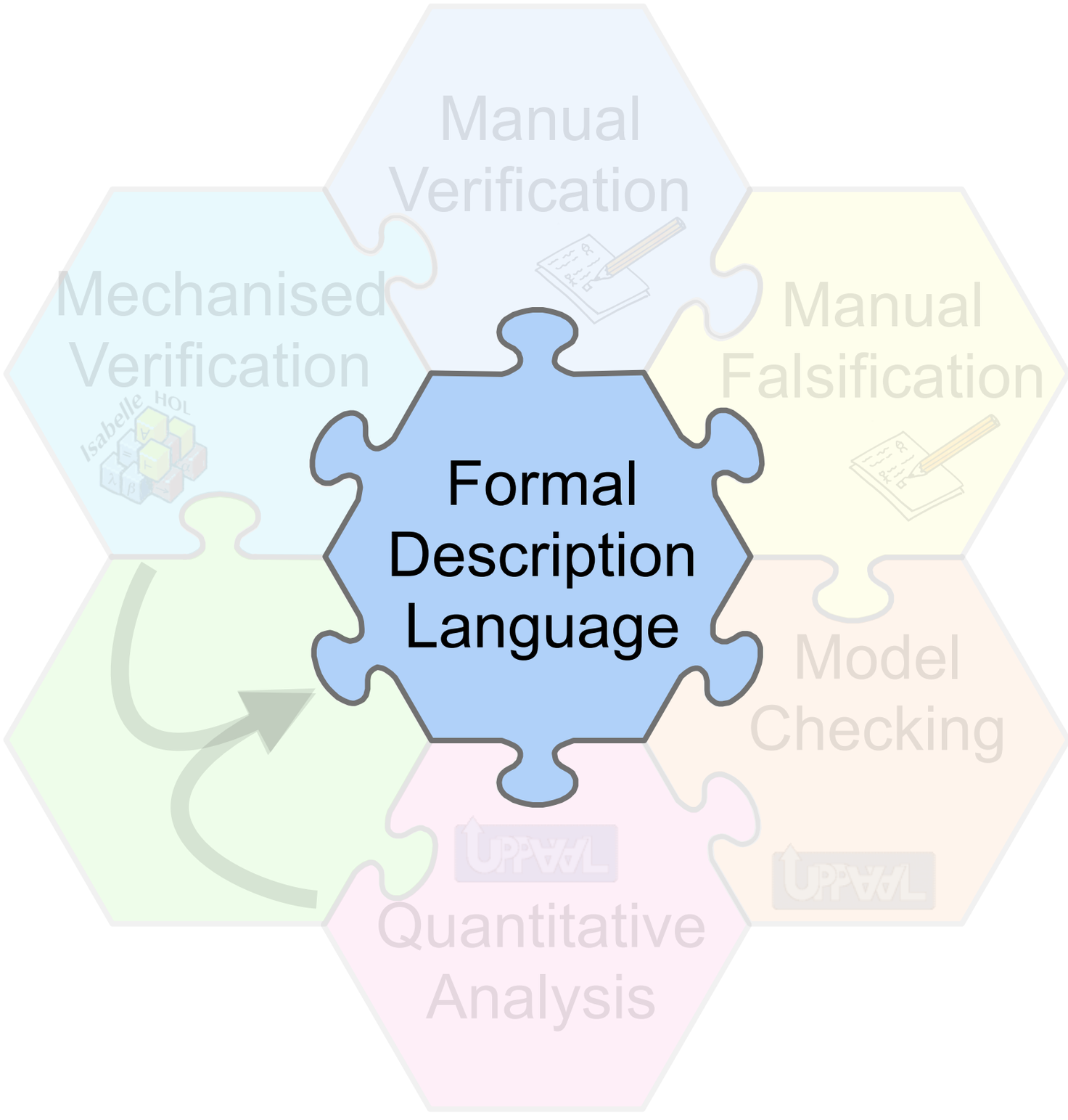
The Big Picture



The Big Picture



The Big Picture



Requirements for Formal Languages

NICTA

- "Formal languages are useful tools for specifying parts of protocols. However, as of today, there exists no well-known language that is able to capture the full syntax and semantics of reasonably rich IETF protocols." [IETF]

- IETF Requirements (for formal methods)
  - relatively easy to extract code
  - complete specification
  - implementation independent
- Additional Requirement
  - easy to use

Desired Constructs

- message sending
  - broadcast
  - unicast
  - groupcast
- data
  - (arbitrary) structures
  - assignments
- operations
  - sequential process
  - choice
  - guards
  - parallelism

Structure of Networks

NICTA

- user (encapsulation)
  - Network as a “cloud”
- collection of nodes
  - connect / disconnect / send / receive
  - “parallel execution” of nodes
- nodes
  - data management
    - data packets, messages, IP addresses ...
  - message management (avoid blocking)
  - core management
    - broadcast / unicast / groupcast ...
  - “parallel execution” of sequential processes

Description Language (Nodes)

NICTA

$X(exp_1, \ldots, \exp_n)$	process calls
$P + Q$	nondeterministic choice
$[\varphi]P$	if-construct (guard)
$[\![\texttt{var} := exp]\!]P$	assignment followed by
$\mathbf{broadcast}(ms).P$	broadcast
$\mathbf{groupcast}(dests, ms).P$	groupcast
$\mathbf{unicast}(dest, ms).P \blacktriangleright Q$	unicast
$\mathbf{send}(ms).P$	send
$\mathbf{receive}(\texttt{msg}).P$	receive
$\mathbf{deliver}(data).P$	deliver

Structural Operational Semantics I

- beyond description language
  - internal state determined by expression and valuation

$$\xi, \mathbf{broadcast}(ms).p \xrightarrow{\mathbf{broadcast}(\xi(ms))} \xi, p$$

$$\xi, \mathbf{groupcast}(dests, ms).p \xrightarrow{\mathbf{groupcast}(\xi(dests),\xi(ms))} \xi, p$$

$$\xi, \mathbf{unicast}(dest, ms).p \blacktriangleright q \xrightarrow{\mathbf{unicast}(\xi(dest),\xi(ms))} \xi, p$$

$$\xi, \mathbf{unicast}(dest, ms).p \blacktriangleright q \xrightarrow{\neg\mathbf{unicast}(\xi(dest))} \xi, q$$

$$\xi, \mathbf{send}(ms).p \xrightarrow{\mathbf{send}(\xi(ms))} \xi, p$$

$$\xi, \mathbf{deliver}(data).p \xrightarrow{\mathbf{deliver}(\xi(data))} \xi, p$$

$$\xi, \mathbf{receive}(\mathtt{msg}).p \xrightarrow{\mathbf{receive}(m)} \xi[\mathtt{msg} := m], p \qquad (\forall m \in \mathtt{MSG})$$

Description Language (2)

$P \lll Q$	parallel operator on nodes

- Do we need more?

$[\varphi]P + [\neg\varphi]Q$	deterministic choice
$P(n) = [\![n := n + 1]\!].P(n)$	loops

Collection of Nodes and Encapsulation

- Node expressions: $M ::= \quad ip : P : R \quad | \quad M \| M$
- Encapsulation: $\mathrm{N} ::= [\mathrm{M}]$

- SOS rules more or less straight-forward

Case Study: AODV

NICTA

$+$ [$(\texttt{oip},\texttt{rreqid}) \notin \texttt{rreqs}$] /* the RREQ is new to this node */
⟦$\texttt{rt} := \texttt{update}(\texttt{rt},(\texttt{oip},\texttt{osn},\texttt{kno},\texttt{val},\texttt{hops}+1,\texttt{sip},\emptyset))$⟧ /* update the route to oip in rt */
⟦$\texttt{rreqs} := \texttt{rreqs} \cup \{(\texttt{oip},\texttt{rreqid})\}$⟧ /* update rreqs by adding (oip, rreqid) */
(
[$\texttt{dip} = \texttt{ip}$] /* this node is the destination node */
⟦$\texttt{sn} := \max(\texttt{sn},\texttt{dsn})$⟧ /* update the sqn of ip */
/* unicast a RREP towards oip of the RREQ */
unicast(nhop(rt,oip),rrep(0,dip,sn,oip,ip)) . AODV(ip,sn,rt,rreqs,store)
▶ /* If the transmission is unsuccessful, a RERR message is generated */
⟦$\texttt{dests} := \{(\texttt{rip},\texttt{inc}(\texttt{sqn}(\texttt{rt},\texttt{rip}))) \mid \texttt{rip} \in \texttt{vD}(\texttt{rt}) \wedge \texttt{nhop}(\texttt{rt},\texttt{rip}) = \texttt{nhop}(\texttt{rt},\texttt{oip})\}$⟧
⟦$\texttt{rt} := \texttt{invalidate}(\texttt{rt},\texttt{dests})$⟧
⟦$\texttt{store} := \texttt{setRRF}(\texttt{store},\texttt{dests})$⟧
⟦$\texttt{pre} := \bigcup\{\texttt{precs}(\texttt{rt},\texttt{rip}) \mid (\texttt{rip},*) \in \texttt{dests}\}$⟧
⟦$\texttt{dests} := \{(\texttt{rip},\texttt{rsn}) \mid (\texttt{rip},\texttt{rsn}) \in \texttt{dests} \wedge \texttt{precs}(\texttt{rt},\texttt{rip}) \neq \emptyset\}$⟧
groupcast(pre,rerr(dests,ip)) . AODV(ip,sn,rt,rreqs,store)
$+$ [$\texttt{dip} \neq \texttt{ip}$] /* this node is not the destination node */
(
[$\texttt{dip} \in \texttt{vD}(\texttt{rt}) \wedge \texttt{dsn} \leq \texttt{sqn}(\texttt{rt},\texttt{dip}) \wedge \texttt{sqnf}(\texttt{rt},\texttt{dip}) = \texttt{kno}$] /* valid route to dip that is fresh enough */
/* update rt by adding precursors */
⟦$\texttt{rt} := \texttt{addpreRT}(\texttt{rt},\texttt{dip},\{\texttt{sip}\})$⟧
⟦$\texttt{rt} := \texttt{addpreRT}(\texttt{rt},\texttt{oip},\{\texttt{nhop}(\texttt{rt},\texttt{dip})\})$⟧
/* unicast a RREP towards the oip of the RREQ */
unicast(nhop(rt,oip),rrep(dhops(rt,dip),dip,sqn(rt,dip),oip,ip)) .

Case Study: AODV

- full specification of AODV (IETF Standard)
- specification details
  - around 5 types and 30 functions
  - around 120 lines of specification
    (in contrast of 40 pages English prose)
- performed rigorous analysis
  (out of scope for today)

Summary

- description language for (wireless networks)
- key features:
  - guarantee broadcast
  - conditional unicast
  - data handling
- case study
  - full concise specification of AODV (RFC 3561) (no time)
- basis for formal analysis

[1] *A Process Algebra for Wireless Mesh Networks.*
A. Fehnker, R. J. van Glabbeek, P. Höfner, A. McIver, M. Portmann, W. L. Tan.
In European Symposium on Programming, LNCS 7211, 295-315 Springer, 2012.

[2] *A Rigorous Analysis of AODV and Its Variants*.
P. Höfner, W. L. Tan, R. J. van Glabbeek, M. Portmann, A. McIver, A. Fehnker.
In Modeling, Analysis and Simulation of Wireless and Mobile Systems, 2012.

[3] *A Process Algebra for Wireless Mesh Networks used for Modelling, Verifying and Analysing AODV*.
A. Fehnker, R. J. van Glabbeek, P. Höfner, A. McIver, M. Portmann, W. L. Tan.
Technical report 5513, NICTA, 2013.

Coming Soon?

NICTA

- Manual Verification using Invariant Proofs
- Model Checking of Network Protocols
- Quantitative Analysis of Network Protocols using Statistical Model Checking
- (Mechanised Verification of Network Protocols)