# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReSpecT at a Glance</td>
<td>3</td>
</tr>
<tr>
<td>What is ReSpecT?</td>
<td>3</td>
</tr>
<tr>
<td>What is ReSpecT for?</td>
<td>3</td>
</tr>
<tr>
<td>Where is ReSpecT?</td>
<td>3</td>
</tr>
</tbody>
</table>
ReSpecT at a Glance

What is ReSpecT?

ReSpecT (Reaction Specification Tuples) is a logic-based coordination language enabling tuple space programming. More precisely, ReSpecT promotes on a tuple-based coordination model, where

- communication is based on logic tuples;
- coordination occur through logic tuple spaces;
- the behaviour of each tuple space is programmable so to embed any computable coordination policy.

In this conceptual framework, ReSpecT has a twofold role:

- as a specification language, ReSpecT makes it possible to catch interaction events and associate them to reactions by means of first-order logic tuples;
- as a reaction language, ReSpecT supports the notion of reaction as a computational activity to be executed locally to a tuple centre in response to interaction events, by providing for both a logic-based syntax for the definition of reactions, and a model for their execution.

What is ReSpecT for?

ReSpecT can be used as a simple yet highly expressive event-driven programming language to program complex process / agent coordination laws outside processes / agents. The more complex is the management of the interaction within a software system the more useful is ReSpecT.

Where is ReSpecT?

Currently, ReSpecT is distributed as a part of the TuCSoN middleware. TuCSoN is a middleware for the coordination of distributed processes, as well as autonomous, intelligent & mobile agents, providing ReSpecT tuple centres as its main coordination abstractions.

Correspondingly, the ReSpecT technology is currently made available as an essential part of the TuCSoN distribution.