



## WILCo - Well Integrated Loosely Coupled



### *Copyright and Disclaimer*

© Boojum Pty Ltd 2003 - 2010 All Rights Reserved

*Boojum Pty Ltd has supplied these materials in good faith, based on information known to us at the time of production. To the extent permitted by law, Boojum Pty Ltd and its employees, agents and consultants will not be held liable for any loss or damage (including indirect, special or consequential loss or damage) arising from the use of, or reliance on, the Information or Products it supplies, whether or not caused by any negligent act or omission. If any law prohibits the exclusion of such liability, Boojum Pty Ltd limits its liability to the extent permitted by law, to the re-supply of the Information or Products.*

### Revision History

Version	Description and reason for change	Author	Date
1	Initial draft	R Maxwell	01 Nov 2009
2	Diagram revised	N Fraser	04 Nov 2009
3	Document formatted for web publication	R Maxwell	16 Feb 2010

# WILCo - Well-Integrated, Loosely Coupled

*Acronym references the radio call sign which means "Will Comply". It's a nickname that we apply at Boojum to our style of system development.*

**Well-Integrated** means that the components work together seamlessly.

**Loosely Coupled** means that if a better component comes on the market, it is a small matter to remove the old one and plug in the new one.

WILCo is a style of software development that we use to advantage our customers. Its aims are:

- lower risk
- better outcomes
- better value for money
- greater flexibility.

There are basically four ways to implement software:

1/ Buy a package, ie a piece of pre-existing software that fits your needs. This is a good approach when you want to do something for which there are well-developed alternatives; word-processing is a good example.

2/ Subscribe to a software service. This is similar to option 1. As web applications become more sophisticated there are more online service options in the market, eg for doing surveys, eg for setting up an online shop.

3/ Custom build a solution. This is appropriate if you have requirements which are not well served by available solutions. A custom application can be part of your competitive advantage, but it can be a sea-anchor if it makes you inflexible and dependent on a single supplier.

4/ WILCo - a WILCo development approach looks at your needs, and then selects pre-existing high-quality components or services to fulfill your needs, and then does custom programming to get the different systems to 'talk' to each other, and to produce additional components for which there is no appropriate pre-existing component. The aim of a WILCo development is to enable your organisation to shape a powerful solution that is tailored to your needs, but which remains agile and cost-effective to enhance. The heart of a WILCo development is the Hub, which must be:

- able to accept input from a wide range of systems (import)
- able to output to a wide range of systems (export)
- built and documented to a high standard, ie the structure, code, and documentation need to be sufficiently clear for new developers to be brought into the mix if desired
- offered to clients under a Free Software or comparable license to ensure that they are not locked into single-supplier dependency.

A WILCo development changes the software development process in a number of ways. Firstly, it puts a greater emphasis on Technical Design and Research. This initial investment is usually repaid by swifter and more sustainable deployment. A WILCo development:

- avoids custom programming where possible, and emphasises the assessment of third-party solutions
- specialises in making disparate elements play harmoniously
- assesses the proposed components in terms of:
  - best fit

- usability
  - functionality
  - quality of code
  - security
  - reliability
  - cost-effectiveness
- prefers Free Software components when they are of equal quality to a proprietary solution, but does not mandate them if a proprietary solution offers superior outcomes
  - concentrates maximum security and data-protection around the Hub, with all input from components being screened and made to conform to a strict data standard
  - enables any component to be replaced easily when a superior component becomes available.

### Schematic overview

