



at the Dutch Food Authority

(NVWA)

case study: the DTV-project



- Customer problem
- Why use & at NVWA?
- Results (so far)
- What did we learn about &?
- Progress in & as a result of the DTV-project



# Customer problem

## Situation

- In jan. 2014, the Minister for Agriculture had made a promise in parliament to solve the problem before Jan 1st, 2015.
- The largest application, SPIN, is in dire need of replacement. That is scheduled for 2017. DTV cannot wait for that.

## Request: DTV (duurzame tijdverantwoording)

- Make and install software for time registration, time processing and time reporting.
- Use a rule-engine for legal correctness.
- Solve the problem sustainably, so that it can be used after 2017.
- Don't change SPIN, for it will be replaced in the near future.
- Make sure it is operational before Jan 1st, 2015.

## Situation:

- Incomplete corporate policy + much discussion w.r.t. rules for time registration.
- August 2014: no team, no tools, and a deadline of Dec 31st...
- Rules are more complicated than anticipated (ARAR, BBRA, WTR, ATW, ...)
- Forced trade with DICTU
- Strictly follow DICTU-standards

## Measures:

- Limit to written rules and legislature (“Show me in writing...”)
  - Design traceably to legal sources
  - Validate rules with experts
  - Design with prototypes
- 
- so: use **&** (as a part of...)



Why use **&** at NVWA?

- **&** provides confidence in legal compliance
  - helps to convince auditors (rekenkamer!)
  - helps to keep discussions relevant (saving time)
- **&** provides prototypes at design time
  - helps to design first-time-right
  - helps to keep users aligned
- **&** generates documentation
  - helps to satisfy DICTU's requirements
  - guarantees that design and prototype correspond
- **&** has been used before by Ordina
  - helps to reassure NVWA
  - makes planning easier

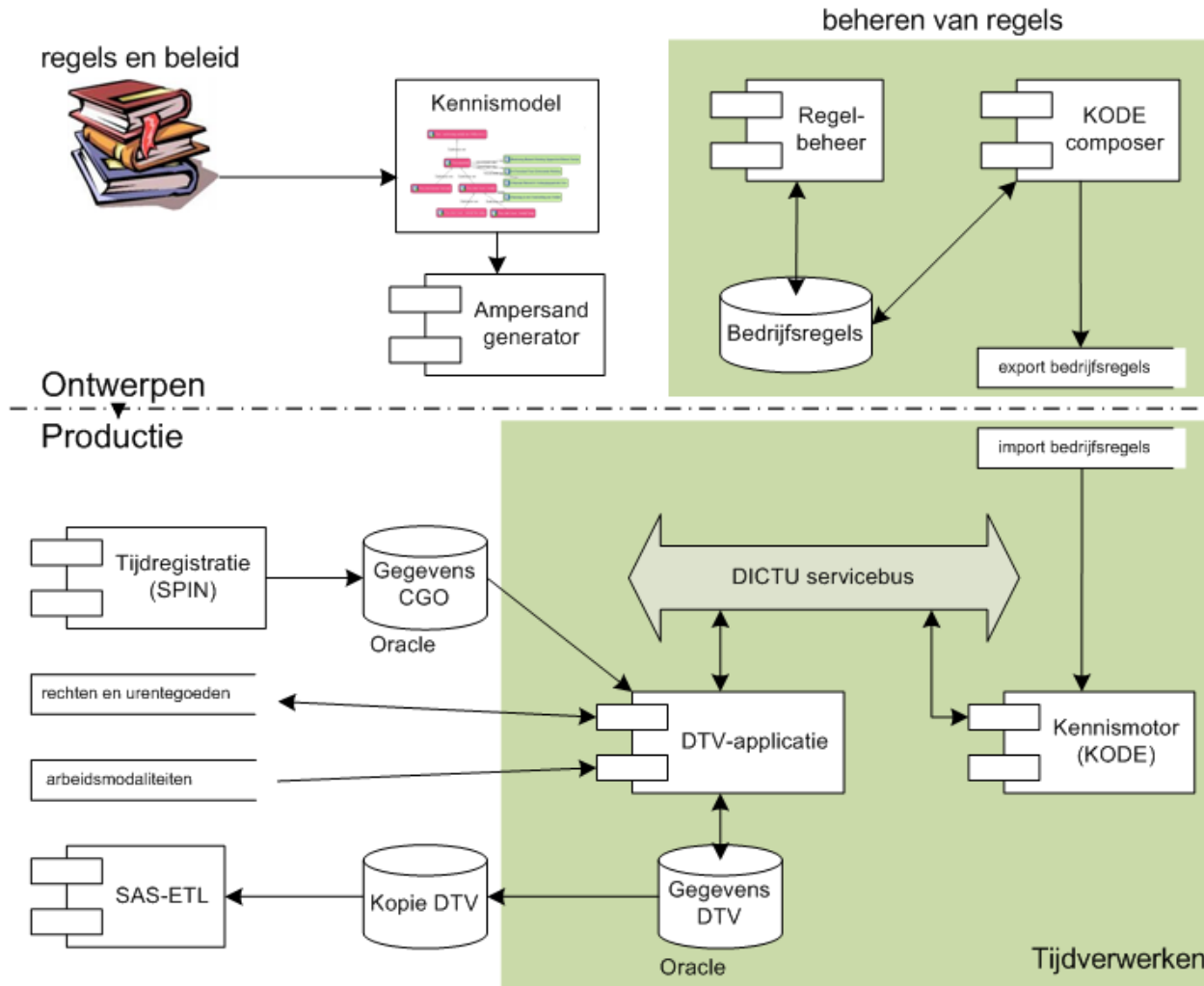




## Results (so far)



- Aug 2014: rule engine selected and installed;
- Sept 2014: elaborated the architecture
- Oct 2014: design in & finished and approval of functional design
- Nov-Dec 2014: time registration and rule engine in production. Time processing in 1<sup>e</sup> delivery.
- On Dec. 29th, the new time registration went live together with the rule engine. (saving the Minister's reputation)



- legal modeling
- landscape consistency shown by formal means
- 3 use cases with 3 prototypes written and verified in october 2014; other use cases follow in summer 2015.
- logical datamodel as input for builders
- prototypes for login and historic reference lists.
- consistency checks for interfaces
- completeness of functional specification
- function point counting
- correctness of chain functionality SPIN – DTV – KODE – DWH
- consistency of message specifications (XML)

- 
- Prototype of historic functionality.



What did we learn about **&**?



# 1. Communication with software engineers

- The functional specification that builders need cannot be generated for the full 100%. We estimate that **&** generates 60% of what is needed.
- software engineers require more detailed specification w.r.t. interfaces.
- The prototypes appear to be very useful, not only to users but to software engineers as well. Example: reference lists
- In spite of the fact that the **&**-prototypes are fully functional database applications , none of the software engineers was curious to look at their inside.

- A major design error in SPIN surfaced in the process of designing and building DTV, due to precision of &.
- Function point counts, generated by &, were used for project estimation purposes.
- The &-model covers the entire architecture. Parts of the same model cover the individuals parts of the architecture. In this way, & provides semantic evidence that the components will work together in practice.

### 3. Communication with users

- Validation of the rules with users proceeded as predicted in the &-manual.
- Decision rules in the rule engine were written in-sync (avg. 1 day delay) with the discussions in the NVWA to clarify time registration policies.
- Prototypes have been used, but not much. Users were very quickly convinced.



**Progress in & as a result of the DTV-project**

# As a result of the DTV-project:

- CRUD-matrices were added to the &-compiler;
- function point counts were refined in the &-compiler;
- the new front-end (by Michiel Stornebrink) got its maiden trip;
- the functional specification generator was improved.



**ICT. MAAR DAN VOOR MENSEN**

[www.ordina.nl](http://www.ordina.nl)