Universiteit Utrecht

[Faculty of **Science**
Information and Computing Sciences]

Relationship Intimacy in Software Ecosystems

A Survey of the Dutch Product Software Industry

Joey van Angeren – Utrecht University
Vincent Blijleven – Utrecht University
Slinger Jansen – Utrecht University

Joey Van Angeren, Vincent Blijleven and Slinger Jansen. Relationship Intimacy in Software Ecosystems: A Survey of the Dutch Software Industry. In the Proceedings of the Conference on Management of Emergent Digital Ecosystems (MEDES 2011)

What is Product Software?

- "A **packaged configuration** of **software components** or a **software-based service**, with auxiliary materials, which is released for and traded in a specific market."

Xu & Brinkkemper, 2005

Supplier Relationships

- As noted from the definition, software products are a **configuration of numerous components**, such as hardware and software components, services and intellectual property

- Often, a software vendor will not develop all of these components in-house, rather there will be a number of **other organizations that supply** these components

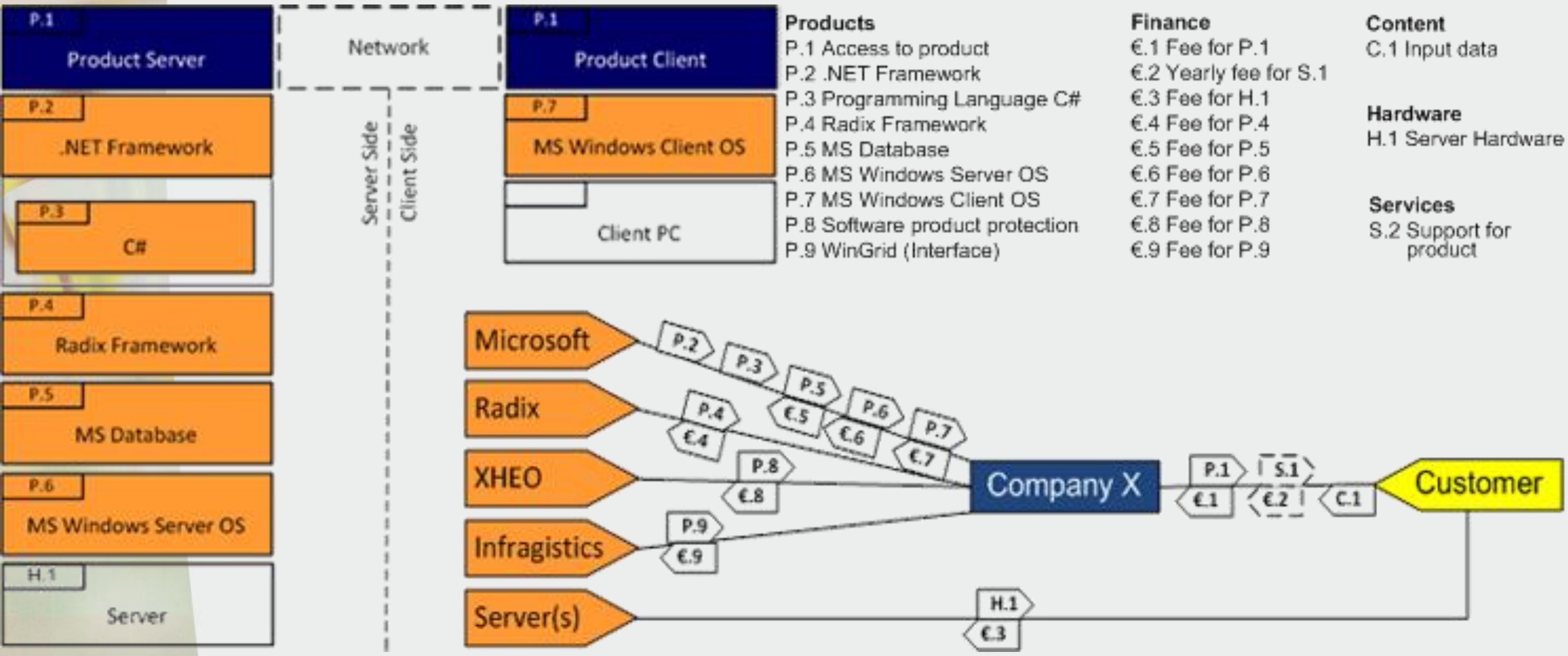
- Software vendors thus **become dependent on suppliers** in order to leverage their products to customers

Why Would We Research Supplier Selection?

- **Researchers' perspective**
  - Little research has been conducted on the scope level of Software Supply Networks within the research domain of software ecosystems
- **Practitioners' perspective**
  - Supplier dependence can have a big impact for a software vendor, for example when the supplier:
    - Decides to alter the product or its license
    - Goes bankrupt or stops business
  - Selecting to join a certain software ecosystem as a customer or partner can bring a wealth of opportunities

Perspectives

- **Software Supply Network (SSN)**
  - Provides insight into first-tier buyer-supplier relationships
- **Product Deployment Context (PDC)**
  - Describes the components of a product and those that are part of its direct running environment in a stack view



Products
P.1 Access to product
P.2 .NET Framework
P.3 Programming Language C#
P.4 Radix Framework
P.5 MS Database
P.6 MS Windows Server OS
P.7 MS Windows Client OS
P.8 Software product protection
P.9 WinGrid (Interface)

Finance
€.1 Fee for P.1
€.2 Yearly fee for S.1
€.3 Fee for H.1
€.4 Fee for P.4
€.5 Fee for P.5
€.6 Fee for P.6
€.7 Fee for P.7
€.8 Fee for P.8
€.9 Fee for P.9

Content
C.1 Input data

Hardware
H.1 Server Hardware

Services
S.2 Support for product

Research Question

- *"How does the perceived level of importance of a component, that is part of a software product, influence supplier selection?"*

Research Method 1/3

- **Data collection**
  - Empirical data from the Dutch product software industry
  - Gathered between September and November 2010
  - Twenty-seven couples of bachelor students during the Product Software course
- **Requirements during the selection process**
  - Number of employees >= 10
  - Registered at the Dutch Chamber of Commerce
- **Two or three meetings, addressing key themes:**
  - General information (employees, products, org. structure)
  - Business models (Osterwalder’s business model canvas)
  - Software ecosystems (SSN, PDC, intimacy suppliers)

Research Method 2/3

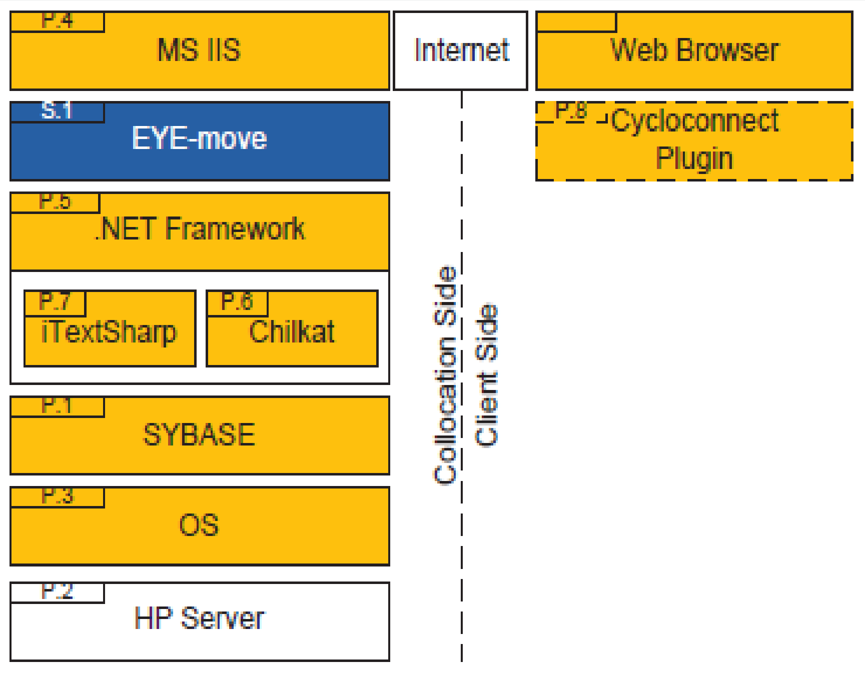
- **Selection Criteria**
  - Inclusion criteria for each contribution to enhance the quality and integrity of the dataset
    - (1) All assignments handed in and accessible
    - (2) Average grade for the entire contribution at least 7,5 (on a scale of 1-10, with 10 being the highest grade)
    - (3) Grade for each assignment at least 7
    - (4) Each assignment entirely executed and complete
    - (5) No duplicates in the dataset
  - 17 out of 27 contributions were included (63%)

Research Method 3/3

- **Data analysis**
  - Quantitative analysis:
    - Contextualizing the dataset by elaborating on **organizational characteristics** such as organizational size and business or delivery models
  - Qualitative analysis:
    - The SSN and PDC, including the table describing the perceived levels of intimacy, was subjected to a **pattern analysis**
  - Notions from existing literature and about the PDC led to the creation of a matrix to classify product components.

Classification Matrix 1/3

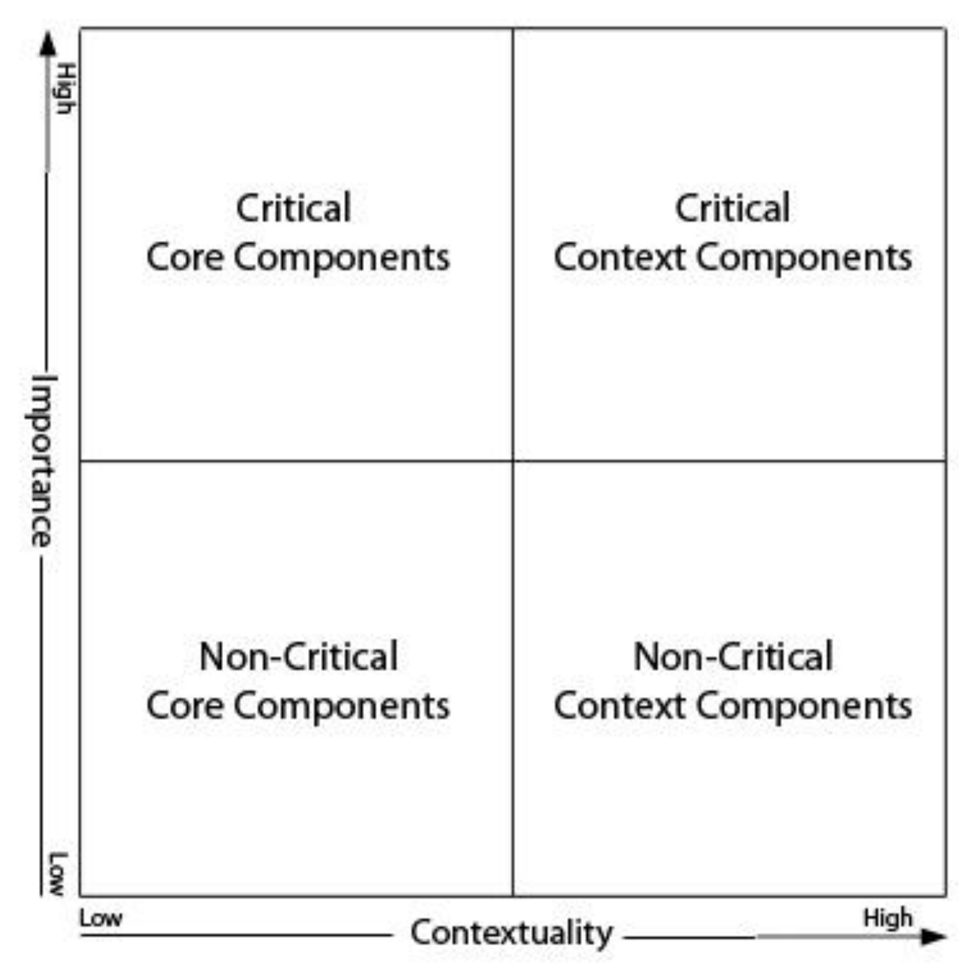
- **Software products**
  - 'Constructed' out of **multiple components**
    - Most relevant components being either hardware or software components
    - Additional services
    - Inclusion of added value
  - Some components obtained from suppliers are more easily interchangeable than others.
    - Easy: interface grid; replacing does not affect the entire product
    - Difficult: migrating to another OS



Boucharas, Jansen & Brinkkemper, 2009

Classification Matrix 2/3

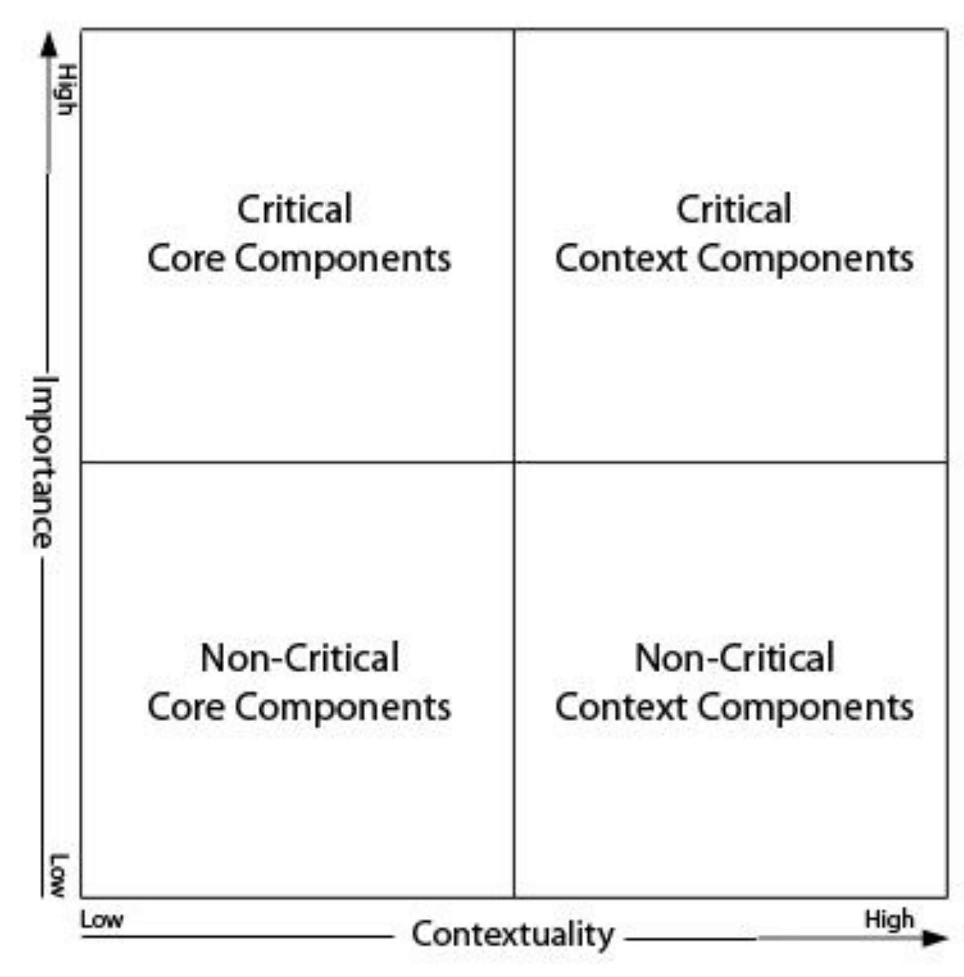
- **Two types of components:**
  - **Core**: fundamental building blocks. Allow the product to be run, without value-added functionalities. ‘Heart’ of the product.
  - **Context**: adds specific value to the product, making it unique. Not necessary to run the product.



High	Critical Core Components	Critical Context Components
Low	Non-Critical Core Components	Non-Critical Context Components

Classification Matrix 3/3

- **Distinction between critical & non-critical components:**
  - **Critical**: interchangeability issues & adds significant value to the overall product
  - **Non-critical**: easily interchangeable with equal functionality

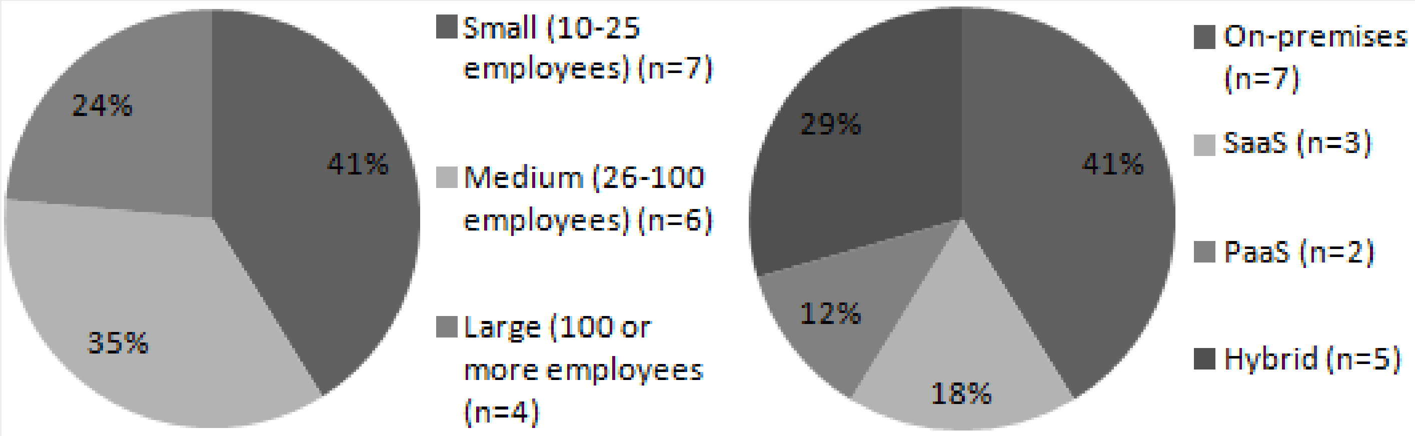


	Low	High
High	Critical Core Components	Critical Context Components
Low	Non-Critical Core Components	Non-Critical Context Components

Conceptualization of the Dataset

- **Three distinct company size categories**
  - Small: 10-25 employees
  - Medium: 26-100 employees
  - Large: > 100 employees

- **Software delivery model**
  - On-premises
  - SaaS
  - PaaS
  - Hybrid



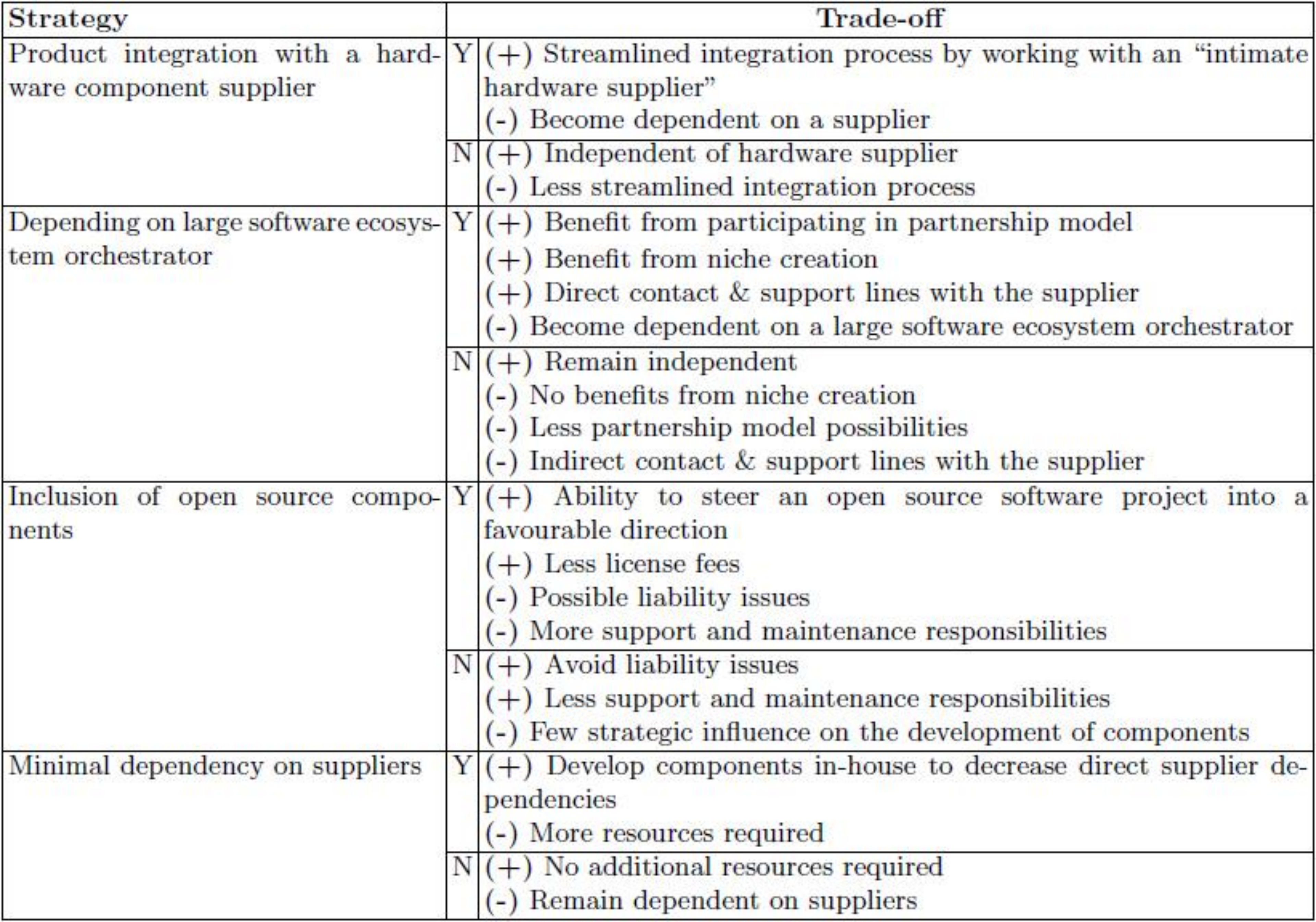
Supplier Selection Strategies 1/2

- **Integrating products with a hardware component supplier**
  - Trade-off:
    - Having a streamlined integration process by working with an ‘intimate hardware supplier’ and thus becoming very dependent on this vendor
    - Having a less streamlined integration process without being dependent on the hardware supplier, since hardware suppliers are generally easily replaceable

- **Dependent on large software ecosystem orchestrators**
  - Benefit from niche creation within the ecosystem.
  - In case of heavily depending on a large software ecosystem as a supplier, it becomes common to join its partnership model.
    - Either become dependent on a large software ecosystem orchestrator with all its benefits
    - Or remain as independent as possible

Supplier Selection Strategies 2/2

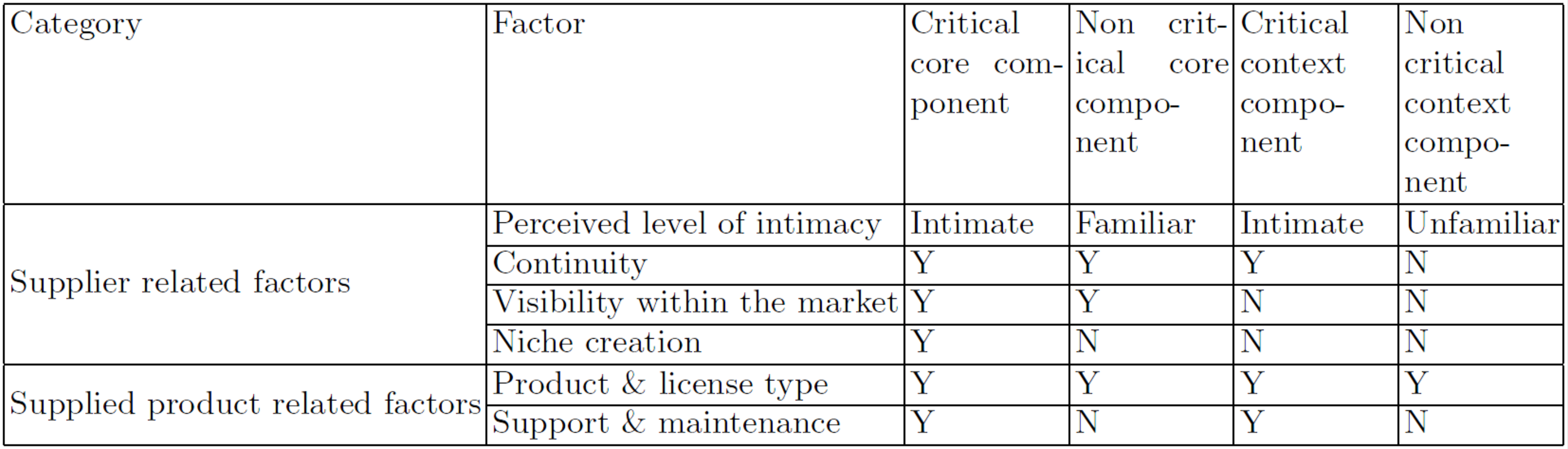
- **Including open source components as alternatives for proprietary components**
  - Trade-off:
    - Include, and contribute to open source to steer the project in a favorable direction
    - Do not include, since more support and maintenance responsibilities will end up with the software vendor rather than its suppliers
- **Minimal dependency strategy**
  - As the organization of the software vendor grows more mature, it becomes attractive to develop components in-house.
  - Trade-off:
    - Either develop components in-house to decrease direct supplier dependencies, requiring more resources
    - Or remain dependent on suppliers, requiring less resources



Strategy	Trade-off	
Product integration with a hardware component supplier	Y	(+) Streamlined integration process by working with an "intimate hardware supplier" (-) Become dependent on a supplier
	N	(+) Independent of hardware supplier (-) Less streamlined integration process
Depending on large software ecosystem orchestrator	Y	(+) Benefit from participating in partnership model (+) Benefit from niche creation (+) Direct contact & support lines with the supplier (-) Become dependent on a large software ecosystem orchestrator
	N	(+) Remain independent (-) No benefits from niche creation (-) Less partnership model possibilities (-) Indirect contact & support lines with the supplier
Inclusion of open source components	Y	(+) Ability to steer an open source software project into a favourable direction (+) Less license fees (-) Possible liability issues (-) More support and maintenance responsibilities
	N	(+) Avoid liability issues (+) Less support and maintenance responsibilities (-) Few strategic influence on the development of components
Minimal dependency on suppliers	Y	(+) Develop components in-house to decrease direct supplier dependencies (-) More resources required
	N	(+) No additional resources required (-) Remain dependent on suppliers

Analysis 1/2

- The **perceived level of intimacy** increases for product with a higher level of perceived importance
- Indicators originating from **ecosystem health measurement** (as proposed by; Den Hartigh, Tol & Visscher, 2006)

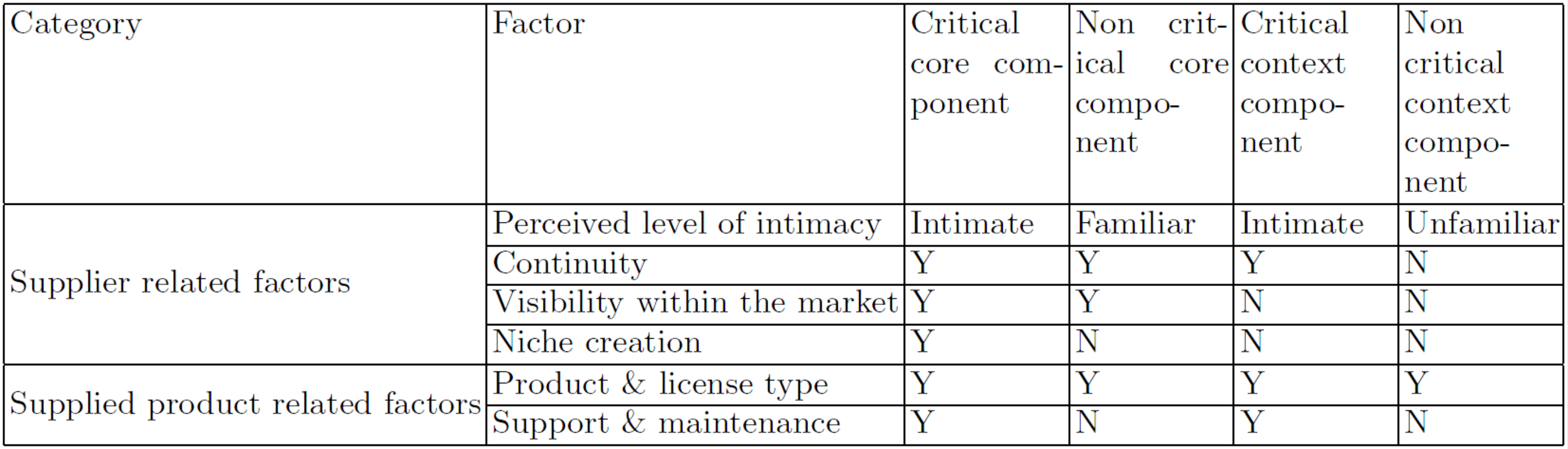


Category	Factor	Critical core component	Non critical core component	Critical context component	Non critical context component
Supplier related factors	Perceived level of intimacy	Intimate	Familiar	Intimate	Unfamiliar
	Continuity	Y	Y	Y	N
	Visibility within the market	Y	Y	N	N
	Niche creation	Y	N	N	N
Supplied product related factors	Product & license type	Y	Y	Y	Y
	Support & maintenance	Y	N	Y	N

Analysis 2/2

- The **type of (open source) license** under which a component is delivered is perceived as vital for (open source) supplier selection

- **Continuous maintenance and support flows** are characteristic for the (product) software industry and therefore prominent in supplier selection



Category	Factor	Critical core component	Non critical core component	Critical context component	Non critical context component
Supplier related factors	Perceived level of intimacy	Intimate	Familiar	Intimate	Unfamiliar
	Continuity	Y	Y	Y	N
	Visibility within the market	Y	Y	N	N
	Niche creation	Y	N	N	N
Supplied product related factors	Product & license type	Y	Y	Y	Y
	Support & maintenance	Y	N	Y	N

Discussion

- Even though the validity of the dataset was enhanced by applying strict selection criteria, the generalization of the results is limited because of the uncommon way in which the data has been gathered
- We cannot state that the SSNs include all the suppliers or that the PDCs contain all relevant components
- Some software vendors may not be aware of some small (open source) components that have, consciously or unconsciously, been incorporated into the leveraged product

Conclusion

- **Components that are part of a Product Deployment Context can be classified by their perceived importance for the end product**

- **Software vendors employ different supplier selection strategies (bringing various trade-offs):**
  - Integrating products with a hardware component supplier
  - Becoming fully reliant on a large software ecosystem
  - Including open source as an alternative for proprietary components
  - Minimal dependency structure

- **The perceived level of importance of a component influences the intimacy supplier relations. Different factors are at play when selecting these:**
  - Factors originating out of ecosystem health measurement (e.g. continuity, visibility within the market, niche creation)
  - Product and license types
  - Maintenance and support flows

Future Research

- Case studies and expert reviews are needed to further evaluate and validate the classification matrix out of a Product Deployment Context and system architectural perspective

- Large sample surveys and case studies of specific cases need to be addressed to increase generalizability for the findings presented

Questions?