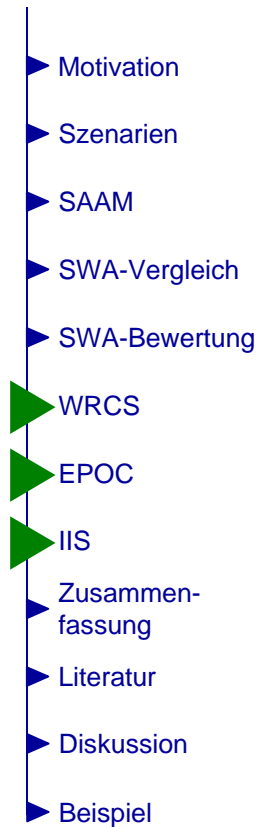




SAAM



- WRCS – *Revision Control System*
- EPOC
- Internet Information Systems (IIS)



SAAM-Anwendung auf ein kommerzielles *Revision Control System* [DeSK 95]

SAAM

- ▶ Motivation
- ▶ Szenarien
- ▶ SAAM
- ▶ SWA-Vergleich
- ▶ SWA-Bewertung
- ▶ WRCS
- ▶ EPOC
- ▶ IIS
- ▶ Zusammenfassung
- ▶ Literatur
- ▶ Diskussion
- ▶ Beispiel

- "WRCS" ist ein erfundener Name für ein (real existierendes) großes, kommerzielles *Revision Control System*
- Vor der Evaluierung existierte keine dokumentierte System-Architektur
- Ziel der Evaluierung war es, die Auswirkungen potenzieller zukünftiger Änderungen (z.B. für eine Portierung) zu bestimmen
- Drei Iterationen
 - Szenario-Entwicklung
 - Architektur-Dokumentation

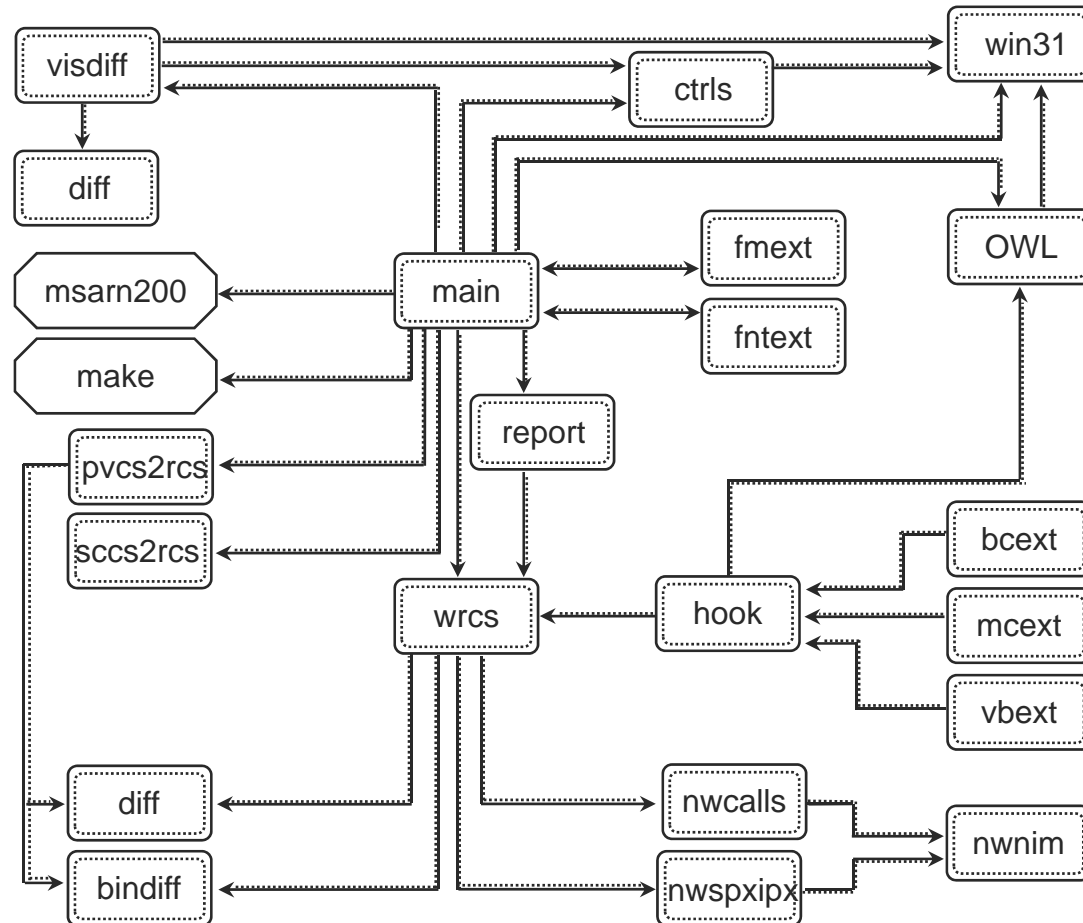
waren notwendig, um eine befriedigende Repräsentation zu erstellen



Architektur-Darstellung von WRCS

SAAM

- Motivation
- Szenarien
- SAAM
- SWA-Vergleich
- SWA-Bewertung
- WRCS**
- EPOC
- IIS
- Zusammenfassung
- Literatur
- Diskussion
- Beispiel



Process

Computational component/object

Data flow and control flow



Beispiel-Szenarien für WRCS...

SAAM

- ▶ Motivation
- ▶ Szenarien
- ▶ SAAM
- ▶ SWA-Vergleich
- ▶ SWA-Bewertung
- ▶ **WRCS**
- ▶ EPOC
- ▶ IIS
- ▶ Zusammenfassung
- ▶ Literatur
- ▶ Diskussion
- ▶ Beispiel

- Benutzer-Szenarien
 - Vergleich von binären Dateien
 - Konfigurieren der Toolbar des Produkts
- Wartungsentwickler
 - Portierung auf ein anderes Betriebssystem
 - Kleinere Änderungen am Benutzer-Interface
- Administrator
 - Zugriffsrechte für ein Projekt ändern
 - Integration in eine neue Entwicklungsumgebung



Szenario-Interaktionen

SAAM

- ▶ Motivation
- ▶ Szenarien
- ▶ SAAM
- ▶ SWA-Vergleich
- ▶ SWA-Bewertung
- ▶ WRCS
- ▶ EPOC
- ▶ IIS
- ▶ Zusammenfassung
- ▶ Literatur
- ▶ Diskussion
- ▶ Beispiel

- Jedes indirekte Szenario verursacht Änderungen in einem oder mehreren Modulen
 - Das kann tabellarisch oder grafisch dargestellt werden
- Auch die *Anzahl* der Szenarien, die jedes Modul betreffen (= Szenario-Interaktionen), kann tabellarisch oder graphisch (*fish-eye view*) dargestellt werden
 - Ein *fish-eye view* signalisiert über die Größe der Module wo (potenzielle) Problembereiche sind



Tabelle: Szenario-Interaktionen

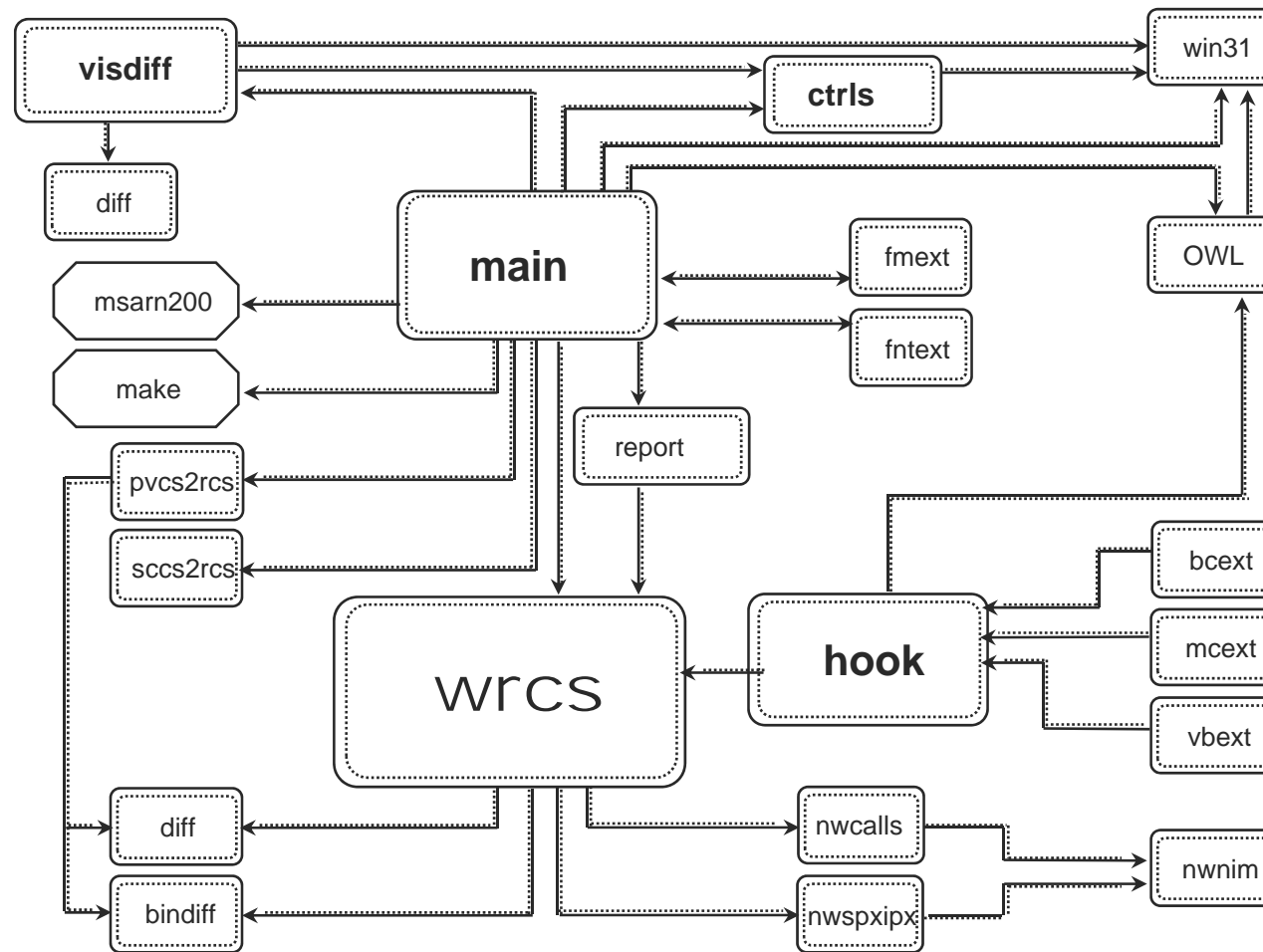
	Modul	Anzahl Änderungen
SAAM	main	4
▶ Motivation	wrcs	7
▶ Szenarien	diff	1
▶ SAAM	bindiff	1
▶ SWA-Vergleich	pvcs2rcs	1
▶ SWA-Bewertung	sccs2rcs	1
▶ WRCS	nwcalls	1
▶ EPOC	nwspxipx	1
▶ IIS	nwnlm	1
▶ Zusammenfassung	hook	4
▶ Literatur	report	1
▶ Diskussion	visdiff	3
▶ Beispiel	ctrls	2



Fish-Eye-View der Szenario-Interaktionen

SAAM

- ▶ Motivation
- ▶ Szenarien
- ▶ SAAM
- ▶ SWA-Vergleich
- ▶ SWA-Bewertung
- ▶ **WRCS**
- ▶ EPOC
- ▶ IIS
- ▶ Zusammenfassung
- ▶ Literatur
- ▶ Diskussion
- ▶ Beispiel





Erfahrungen: Granularität der Architekturbeschreibung

SAAM

- ▶ Motivation
- ▶ Szenarien
- ▶ SAAM
- ▶ SWA-Vergleich
- ▶ SWA-Bewertung
- ▶ WRCS
- ▶ EPOC
- ▶ IIS
- ▶ Zusammenfassung
- ▶ Literatur
- ▶ Diskussion
- ▶ Beispiel

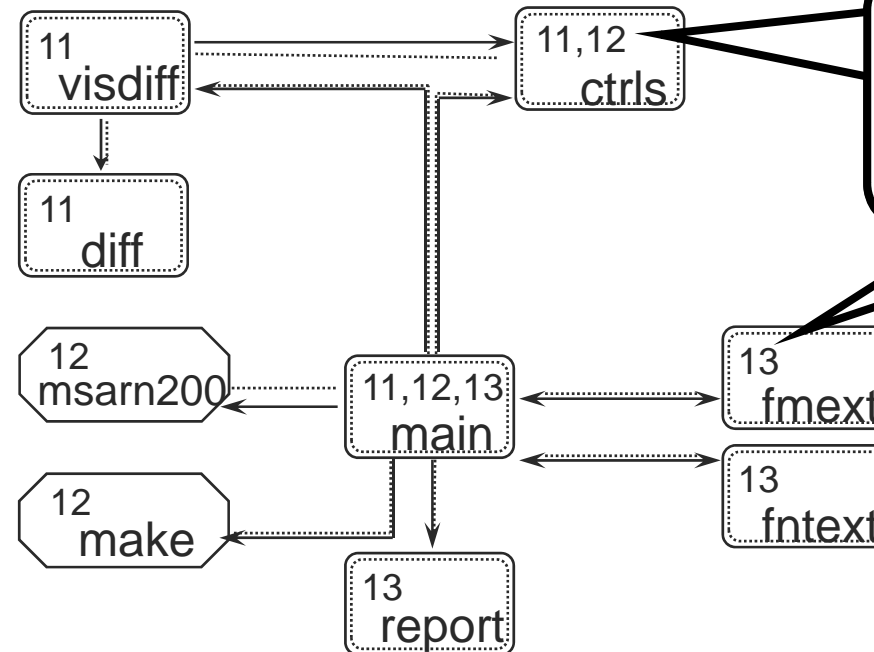
- Der nötige Detailgrad der Architekturbeschreibung wird durch die gewählten Szenarien bestimmt
 - Zur Illustration s. nächste Folie:
ursprünglicher Vorschlag des Architekten
- Nummern bezeichnen die indirekten Szenarien, welche die jeweiligen Komponenten betreffen



Original-Darstellung der WRCS-Architektur; Szenario-Interaktionen mit "main"

SAAM

- Motivation
- Szenarien
- SAAM
- SWA-Vergleich
- SWA-Bewertung
- WRCS**
- EPOC
- IIS
- Zusammenfassung
- Literatur
- Diskussion
- Beispiel



Möglichkeiten bzgl. "main"

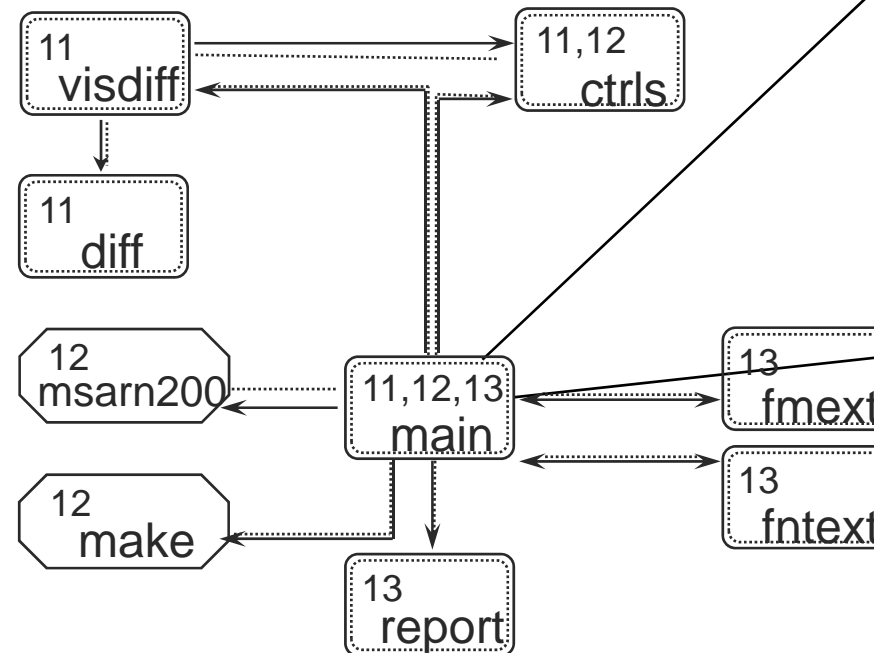
- Die Szenarien 11, 12 und 13 gehören zur selben Klasse
- Die Szenarien gehören zu verschiedenen Klassen und "main" *kann nicht* weiter unterteilt werden
- Die Szenarien gehören zu verschiedenen Klassen und "main" *kann* weiter unterteilt werden



Original-Darstellung der WRCS-Architektur; Szenario-Interaktionen mit "main"

SAAM

- Motivation
- Szenarien
- SAAM
- SWA-Vergleich
- SWA-Bewertung
- WRCS**
- EPOC
- IIS
- Zusammenfassung
- Literatur
- Diskussion
- Beispiel



Möglichkeiten bzgl. "main"

- Die Szenarien 11, 12 und 13 gehören zur selben Klasse
- Die Szenarien gehören zu verschiedenen Klassen und "main" *kann nicht* weiter unterteilt werden
- Die Szenarien gehören zu verschiedenen Klassen und "main" *kann* weiter unterteilt werden



Erkenntnisse für WRCS

SAAM

- ▶ Motivation
- ▶ Szenarien
- ▶ SAAM
- ▶ SWA-Vergleich
- ▶ SWA-Bewertung
- ▶ WRCS
- ▶ EPOC
- ▶ IIS
- ▶ Zusammenfassung
- ▶ Literatur
- ▶ Diskussion
- ▶ Beispiel

- Es wurden ernste Einschränkungen der gewünschten Portabilität und Modifizierbarkeit identifiziert
Empfehlung: "*major system redesign*"
- Die Evaluierung wurde von den Entwicklern unterschiedlich empfunden
 - Senior Developers/Managers fanden sie wichtig und hilfreich
 - Andere Entwickler betrachteten sie als eine akademische Übung
- SAAM ermöglichte Einblicke in Fähigkeiten und Beschränkungen, die sonst schwierig zu entdecken waren
- Die Ergebnisse wurden *mit nur sehr oberflächlichem Wissen* über die Interna von WRCS ermittelt!



EPOC Assessment [Talgren 01]

SAAM

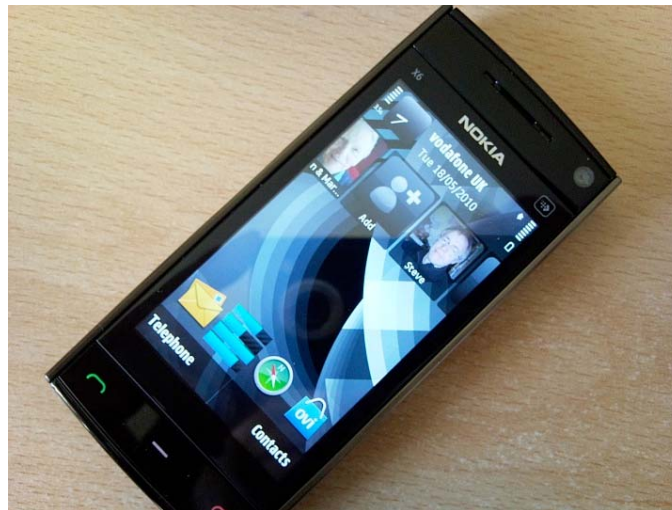
- ▶ Motivation
- ▶ Szenarien
- ▶ SAAM
- ▶ SWA-Vergleich
- ▶ SWA-Bewertung
- ▶ WRCS
- ▶ EPOC
- ▶ IIS
- ▶ Zusammenfassung
- ▶ Literatur
- ▶ Diskussion

ESAPS

Folie 55

Vorlesung SWA | Prof. Dr. Peter Knauber | HS MA, SS 2013

- Kontext der Evaluierung: ESAPS-Projekt
- EPOC: Betriebssystem für Handhelds, Smartphones etc.
 - Früher: Psion Software (Release 1 – 5)
 - Mittlerweile: Symbian (ab Release 6)
 - Open Source seit 8/2009: Release Symbian^2
- Name (je nachdem, ob man Manager oder Entwickler fragt)
 - Abgeleitet von "Epoche" / "epochal"
 - Abkürzung von *Electronic Piece of Cheese*



Nokia X6 16/32 GB (5/2010)



SAAM

- ▶ Motivation
- ▶ Szenarien
- ▶ SAAM
- ▶ SWA-Vergleich
- ▶ SWA-Bewertung
- ▶ WRCS
- ▶ EPOC
- ▶ IIS
- ▶ Zusammenfassung
- ▶ Literatur
- ▶ Diskussion
- ▶ Beispiel

1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | next > last >

 Nokia N8 Symbian^3 Learn more	 Nokia E5 S60 3rd Edition FP2 Learn more	 Nokia C5 S60 3rd Edition FP2 Learn more	 Nokia 5230 Nuron Symbian^1 Learn more
 Sony Ericsson Vivaz Pro Symbian^1 Learn more	 Sony Ericsson Vivaz Symbian^1 Learn more	 Fujitsu docomo SMART series F-03B MOAP(S) v6 Learn more	 Samsung i8910 HD Symbian^1 Learn more
 Nokia E72 S60 3rd Edition FP2 Learn more	 Nokia X5 S60 3rd Edition FP2 Learn more	 Sharp docomo STYLE series SH-06B MOAP(S) v6 Learn more	 Nokia C6 Symbian^1 Learn more
 Fujitsu docomo PRIME series F-04B MOAP(S) v6 Learn more	 Sharp docomo PRO series SH-03B MOAP(S) v6 Learn more	 Sharp docomo STYLE series SH-05B MOAP(S) v6 Learn more	 Sharp docomo STYLE series SH-04B MOAP(S) v6 Learn more

Visit us on ideas.symbian.org



SAAM

- ▶ Motivation
- ▶ Szenarien
- ▶ SAAM
- ▶ SWA-Vergleich
- ▶ SWA-Bewertung
- ▶ WRCS
- ▶ EPOC
- ▶ IIS
- ▶ Zusammenfassung
- ▶ Literatur
- ▶ Diskussion

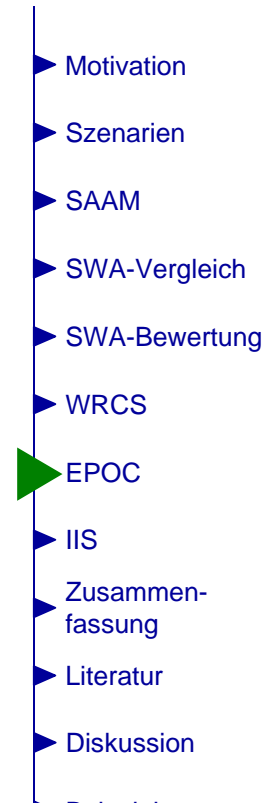


- Assessment im Auftrag von Nokia und Ericsson, Fokus:
 - Technische Grenzen erforschen
 - Adaptierbarkeit für zukünftige Anforderungen erforschen
 - Eignung für Produktlinien-Entwicklung ausloten
 - Erfahrungen mit dem Assessment-Prozess sammeln
- Herausfinden, was Symbian tun muss/müsste und was die Lizenznehmer selbst tun können
- Problem: Pläne für die EPOC-Weiterentwicklung und Roadmaps der Lizenznehmer sind *vertraulich!*



SAAM-Teilnehmer

SAAM



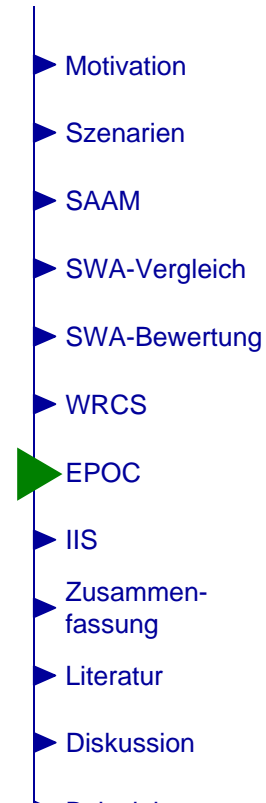
- Moderator (*Facilitator*)
- Zwei Protokollführer
 - einer inhaltlich
 - einer prozess-bezogen
- EPOC-Experten
- EPOC-Benutzer (Nokia, Ericsson)

Dauer: 5 – 6 Stunden





SAAM



- Szenarien waren aus zwei konkreten Produkt-Roadmaps abgeleitet
- Zwei getrennte Szenarien-Gruppen wurden zu 16 initialen Szenarien weiterentwickelt
 - konkretisiert
 - zusammengefasst
 - aufgeteilt
 - verworfen

Ergebnis

- Acht besonders wichtige Szenarien
- Klassifikation:
 - direkt
 - semi-direkt (von zukünftig geplanten EPOC-Releases unterstützt)
 - indirekt





SAAM

- ▶ Motivation
- ▶ Szenarien
- ▶ SAAM
- ▶ SWA-Vergleich
- ▶ SWA-Bewertung
- ▶ WRCS
- ▶ EPOC
- ▶ IIS
- ▶ Zusammenfassung
- ▶ Literatur
- ▶ Diskussion
- ▶ Beispiel

Frage:

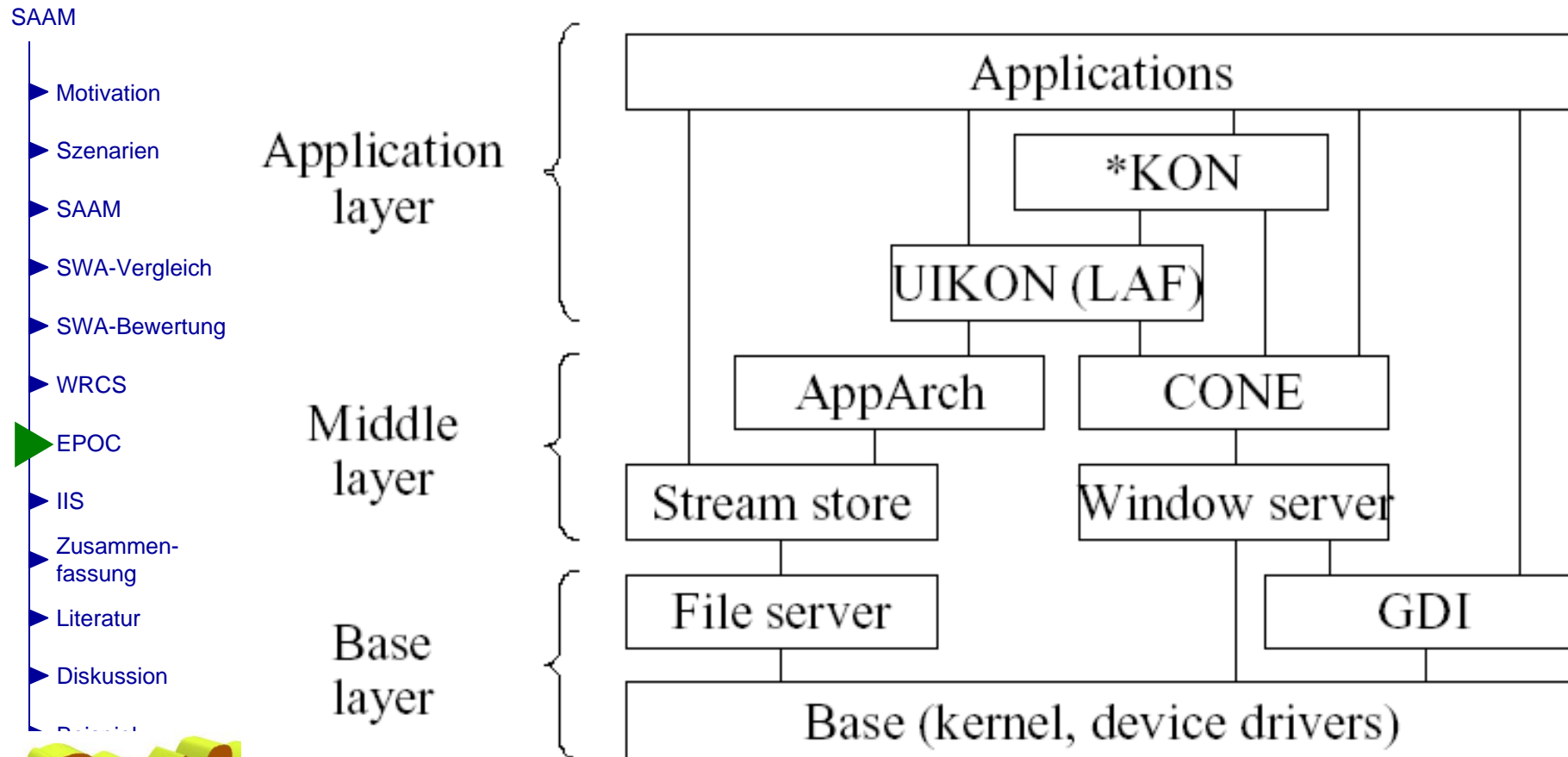
Welcher Architekturstil für Betriebssysteme?

Warum?





EPOC-Architektur





Szenarien: Authentication and authorisation

SAAM

- ▶ Motivation
- ▶ Szenarien
- ▶ SAAM
- ▶ SWA-Vergleich
- ▶ SWA-Bewertung
- ▶ WRCS
- ▶ EPOC
- ▶ IIS
- ▶ Zusammenfassung
- ▶ Literatur
- ▶ Diskussion



- Description
The user of the device wants to make a *payment* with the mobile device. What mechanisms exist to support this and what mechanisms exist for creating proprietary solutions?
- Evaluation
This is not supported directly, and there are no plans to support this in the future. SSL and WAP is supported, so it is possible to create custom solutions that conduct authentication on the network side.



Szenarien: User profiles

SAAM

- ▶ Motivation
- ▶ Szenarien
- ▶ SAAM
- ▶ SWA-Vergleich
- ▶ SWA-Bewertung
- ▶ WRCS
- ▶ EPOC
- ▶ IIS
- ▶ Zusammenfassung
- ▶ Literatur
- ▶ Diskussion



- Description
Is there support for *multiple users with different user rights*, or for the same person in a different role?
For instance, a user might want to store work and home mobile phone numbers separately, so that the work numbers could be updated automatically.
- Evaluation
This is not currently directly supported and there are not public plans to support this in the future. Applications can implement user profiles themselves, but this would be tricky to make secure, as the file system does not support different user rights.

One solution would be to store the user profile on a compact flash that the user would insert whenever starting to use an application. However, this also needs to be done separately for each application.

As for the built-in applications, support for multiple mailboxes is currently being implemented. Apart from this, the standard applications do not support multiple users. EPOC is inherently a single-user operating system.



Szenarien: New I/O models

SAAM

- ▶ Motivation
- ▶ Szenarien
- ▶ SAAM
- ▶ SWA-Vergleich
- ▶ SWA-Bewertung
- ▶ WRCS
- ▶ EPOC
- ▶ IIS
- ▶ Zusammenfassung
- ▶ Literatur
- ▶ Diskussion



- Description
What support is there for creating new I/O models?
Concretely, how difficult would it be to create the *user interface* for a device with *two separate keypads and displays*? An example of such a device is the Nokia Communicator (9110).
- Evaluation
Creating a new Device Family Reference Design takes typically 15 person years to specify and 1.5 years in calendar time. Symbian will create the new DFRD.

As for two keypads, all key events are propagated through the device driver to Window Server (WSERV) and then to UIKON. For handwriting recognition, Symbian created an interception mechanism between WSERV and UIKON. The same mechanism could be used when there are two keypads. The application would not notice any difference.

The case of two different displays has not been considered. It requires modifications to WSERV but the modifications would not be very difficult. WSERV does its work by writing to display memory. The two displays could use adjacent pieces of display memory. A licensee cannot do the modifications.

Creating applications that adapt to different display sizes is possible, as Ericsson has done that with the R320 device.



Szenarien: Real-time support

SAAM

- ▶ Motivation
- ▶ Szenarien
- ▶ SAAM
- ▶ SWA-Vergleich
- ▶ SWA-Bewertung
- ▶ WRCS
- ▶ EPOC
- ▶ IIS
- ▶ Zusammenfassung
- ▶ Literatur
- ▶ Diskussion



- Description
Many envisioned applications of EPOC devices, such as telephony and multimedia, require *guaranteed response times* from the operating system. How is it possible to create real-time applications on EPOC?
- Evaluation
EPOC is not currently a real-time operating system. This is because of design decisions made to balance performance against determinacy. For instance, the practice of using a single page directory makes context switching very fast, but since all processes use consecutive memory chunks, relocation can have large overhead.

One solution at the systems engineering level is to use two processor cores, one for real-time applications and the other for the user interface and applications. This obviously makes the chip larger.

As an alternative to adding real-time support to EPOC, the licensees could use low-level mechanisms to bypass the operating system. For instance, one could add a non-preemptable thread with full hardware access running at the highest priority. It is not clear how difficult this would be or how much knowledge of EPOC internals it would require.



Weitere Szenarien

SAAM

- ▶ Motivation
- ▶ Szenarien
- ▶ SAAM
- ▶ SWA-Vergleich
- ▶ SWA-Bewertung
- ▶ WRCS
- ▶ EPOC
- ▶ IIS
- ▶ Zusammenfassung
- ▶ Literatur
- ▶ Diskussion



- Data security
Data security requires mechanisms to run *applications with limited rights*. The sensitive parts in an EPOC device are the user's data and the telephony network interface. What mechanisms exist to limit the rights of an application?
- Dynamic software upgrade
How easy is it to *upgrade the software* on a device?
- Fault tolerance
What will happen if the device suddenly *loses power or runs out of memory*? What features exist for high availability?
- Low-end phone
Is it possible to make an EPOC phone that would cost 250 – 300 euros and would have little resources (say 8-12 Mbytes of RAM and a 100 MHz processor). This phone would have less resources than the currently envisioned EPOC devices, but could be an *attractive addition to a product line*.



Nicht-evaluierte Szenarien-Themen

SAAM

- ▶ Motivation
- ▶ Szenarien
- ▶ SAAM
- ▶ SWA-Vergleich
- ▶ SWA-Bewertung
- ▶ WRCS
- ▶ EPOC
- ▶ IIS
- ▶ Zusammenfassung
- ▶ Literatur
- ▶ Diskussion



- Multiple IP addresses
- Multiple communication channels active at the same time
- Multimedia
- Country and language adaptations
- Security
- External storage (Compact-flash, Memory-stick etc.)
- Upgradeable software
- Power management
- Components and reuse
- Open platform

Development environment





Fallstudie: Internet Information Systems (IIS) [KBAC95]

SAAM

- ▶ Motivation
- ▶ Szenarien
- ▶ SAAM
- ▶ SWA-Vergleich
- ▶ SWA-Bewertung
- ▶ WRCS
- ▶ EPOC
- ▶ IIS
- ▶ Zusammenfassung
- ▶ Literatur
- ▶ Diskussion
- ▶ Beispiel

- World-Wide Web (WWW)
 - Ist ja wohl hinreichend bekannt...
- WAIS (Wide Area Information Servers)
 - Network Publishing System
 - Abfragen mischen natürliche Sprache mit Boole'schen Ausdrücken
- Prospero
 - Dateien anhand von lokal definierten Views übers Internet ansehen
 - Hauptfokus: Organisation (nicht: Suche) von Informationen



Ablauf des Vergleichs

SAAM

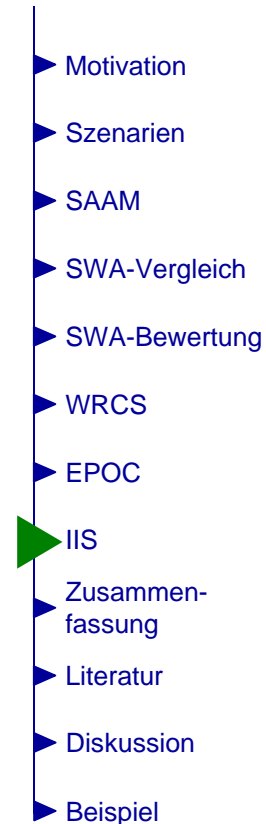
- ▶ Motivation
- ▶ Szenarien
- ▶ SAAM
- ▶ SWA-Vergleich
- ▶ SWA-Bewertung
- ▶ WRCS
- ▶ EPOC
- ▶ IIS
- ▶ Zusammenfassung
- ▶ Literatur
- ▶ Diskussion
- ▶ Beispiel

1. Rollen identifizieren
2. Szenarien aus Sicht der Rollen erstellen
3. Referenz-Funktionalität erarbeiten und klassifizieren
4. Drei Systemarchitekturen in einer einheitlichen Notation darstellen
5. Referenz-Funktionalität auf die Systemkomponenten abbilden
6. Prüfen, welche Szenarien direkt unterstützt werden
7. Indirekte Szenarien anhand der Abbildungen der involvierten Funktionalität bewerten



1. Stakeholder, Stakeholder-Rollen

SAAM



- Informationskonsument
benutzt ein IIS, um im Internet nach Informationen zu suchen
- Informationsproduzent
macht Information im Internet verfügbar, als Stand-Alone-Repository (eigene *web site*) oder als Beitrag zu einem existierenden Repository (einer existierenden *web site*)
- Entwickler
modifiziert einen Teil der Infrastruktur eines IIS



2. Szenarien

SAAM

- ▶ Motivation
- ▶ Szenarien
- ▶ SAAM
- ▶ SWA-Vergleich
- ▶ SWA-Bewertung
- ▶ WRCS
- ▶ EPOC
- ▶ IIS
- ▶ Zusammenfassung
- ▶ Literatur
- ▶ Diskussion
- ▶ Beispiel

- a) Suche nach einer bestimmten Information (klares Ziel)
- b) Unspezifische Suche nach Informationen (*browsing*, kein spezielles Ziel)
- c) Information zu einem existierenden Repository hinzufügen
- d) Ein Repository eines existierenden Typs an einer neuen Stelle erstellen
- e) Ein Repository eines existierenden Typs an einer existierenden Stelle erstellen
- f) Eine neue Sicht auf existierende Informationen einrichten
- g) Informationen innerhalb eines Repositories oder zu einem anderen Repository verschieben (zwecks Reorganisation)
- h) Einen neuen Zugriffsmechanismus hinzufügen
- i) Einen neuen Informationstyp hinzufügen



3. **Referenz-Funktionalität**, entspricht **nicht** Komponenten!

SAAM

- ▶ Motivation
- ▶ Szenarien
- ▶ SAAM
- ▶ SWA-Vergleich
- ▶ SWA-Bewertung
- ▶ WRCS
- ▶ EPOC
- ▶ IIS
- ▶ Zusammenfassung
- ▶ Literatur
- ▶ Diskussion
- ▶ Beispiel

- UI User Interface
- IA Information Access Engine (Formulieren von Anfragen)
- IX Index (für die Suche)
- DR Data Repository
- DW Database Wrapper (einheitlicher Zugriff auf heterogene Repositories)
- LD Logical Database (*view* auf Daten)
- CM Communication Manager (z.B. TCP/IP)
- NR Name Resolver (abstrahiert mittels Namen von Repository-Orten)
- MV Move Utility (aktualisiert die Tabellen des NR)



2. Klassifikation der Szenarien (Erwartung bzgl. der Referenz-Funktionalität!)

SAAM

- ▶ Motivation
- ▶ Szenarien
- ▶ SAAM
- ▶ SWA-Vergleich
- ▶ SWA-Bewertu
- ▶ WRCS
- ▶ EPOC
- ▶ IIS
- ▶ Zusammenfassung
- ▶ Literatur
- ▶ Diskussion
- ▶ Beispiel

Task Scenario	Classification	Functional coupling
search for information	direct	
browse for information	direct	
add information to an existing repository	direct	
add repository (of an existing type) at a new location	direct	
add new repository (of an existing type) at an existing location	direct	
add new view of pre-existing information	direct	
move information (for reorganization purposes)	direct	
add new type of data repository	indirect	DW, DR
add new information access engine	indirect	UI, IA, DW

Erklärung
s. nächste Folie



3. Referenz-Funktionalität

SAAM

- ▶ Motivation
- ▶ Szenarien
- ▶ SAAM
- ▶ SWA-Vergleich
- ▶ SWA-Bewertung
- ▶ WRCS
- ▶ EPOC
- ▶ IIS
- ▶ Zusammenfassung
- ▶ Literatur
- ▶ Diskussion
- ▶ Beispiel

- UI User Interface
- IA Information Access Engine (Formulieren von Anfragen)
- IX Index (für die Suche)
- DR Data Repository
- DW Database Wrapper (einheitlicher Zugriff auf heterogene Repositories)
- LD Logical Database (*view* auf Daten)
- CM Communication Manager (z.B. TCP/IP)
- NR Name Resolver (abstrahiert mittels Namen von Repository-Orten)

add new type of data repository	indirect	DW, DR
add new information access engine	indirect	UI, IA, DW



4. Architektur-Notation

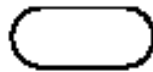
SAAM

- Motivation
- Szenarien
- SAAM
- SWA-Vergleich
- SWA-Bewertung
- WRCS
- EPOC
- IIS**
- Zusammenfassung
- Literatur
- Diskussion
- Beispiel

Components



Process



Computational Component

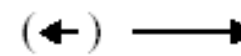


Passive Data Repository



Active Data Repository

Connections



Uni-/Bi-directional Control Flow



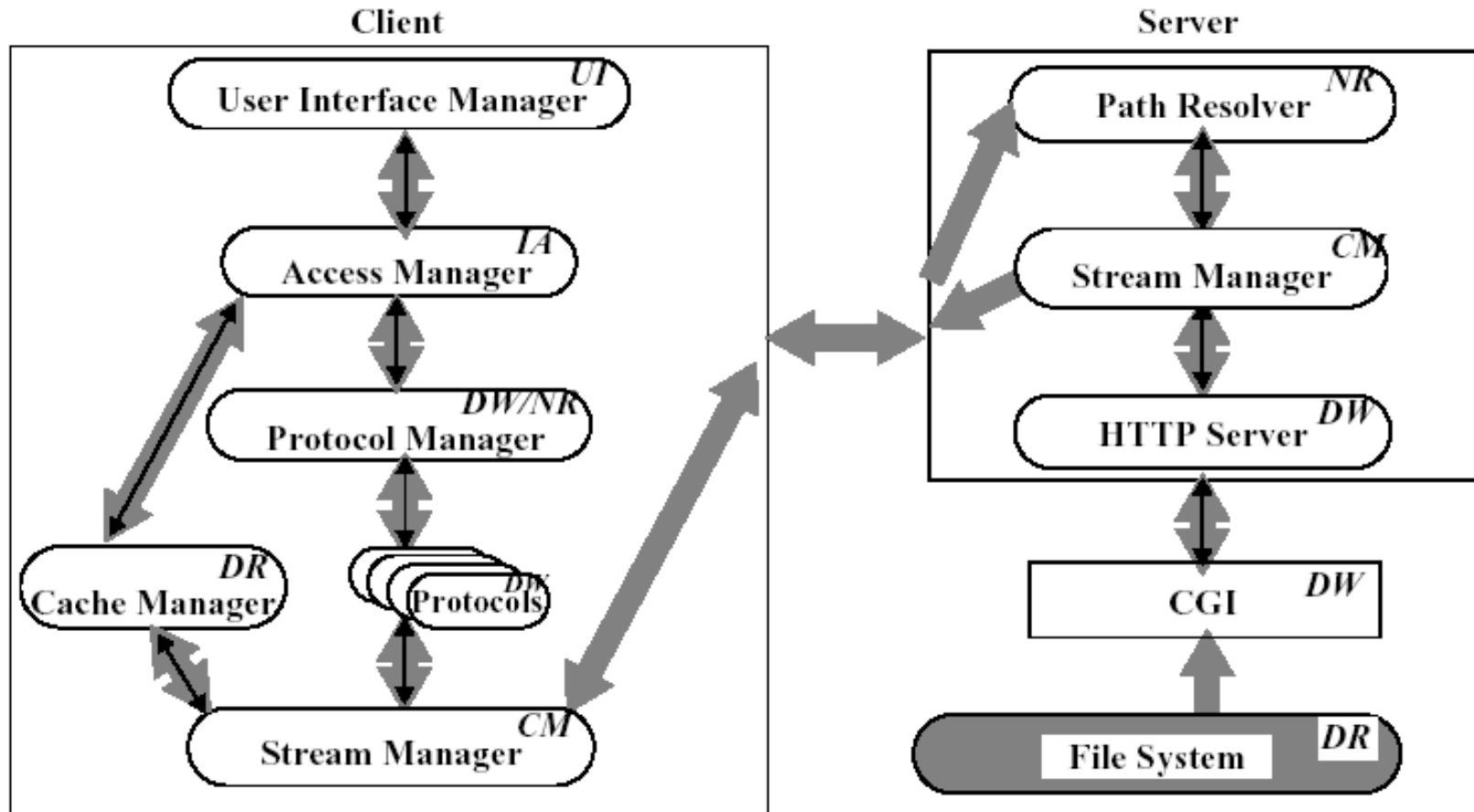
Uni-/Bi-directional Data Flow



4.+5. Typische WWW-Architektur

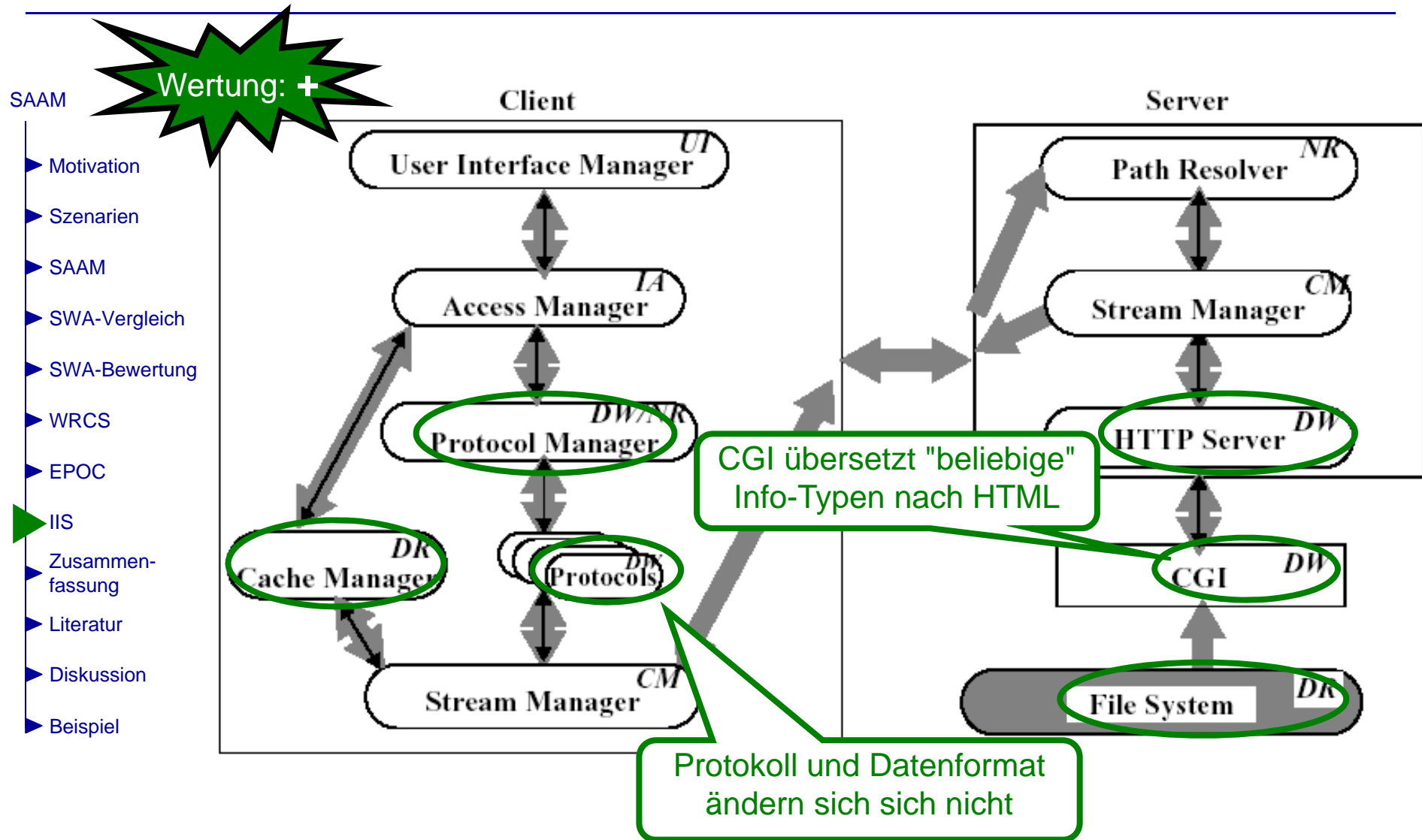
SAAM

- ▶ Motivation
- ▶ Szenarien
- ▶ SAAM
- ▶ SWA-Vergleich
- ▶ SWA-Bewertung
- ▶ WRCS
- ▶ EPOC
- ▶ IIS
- ▶ Zusammenfassung
- ▶ Literatur
- ▶ Diskussion
- ▶ Beispiel



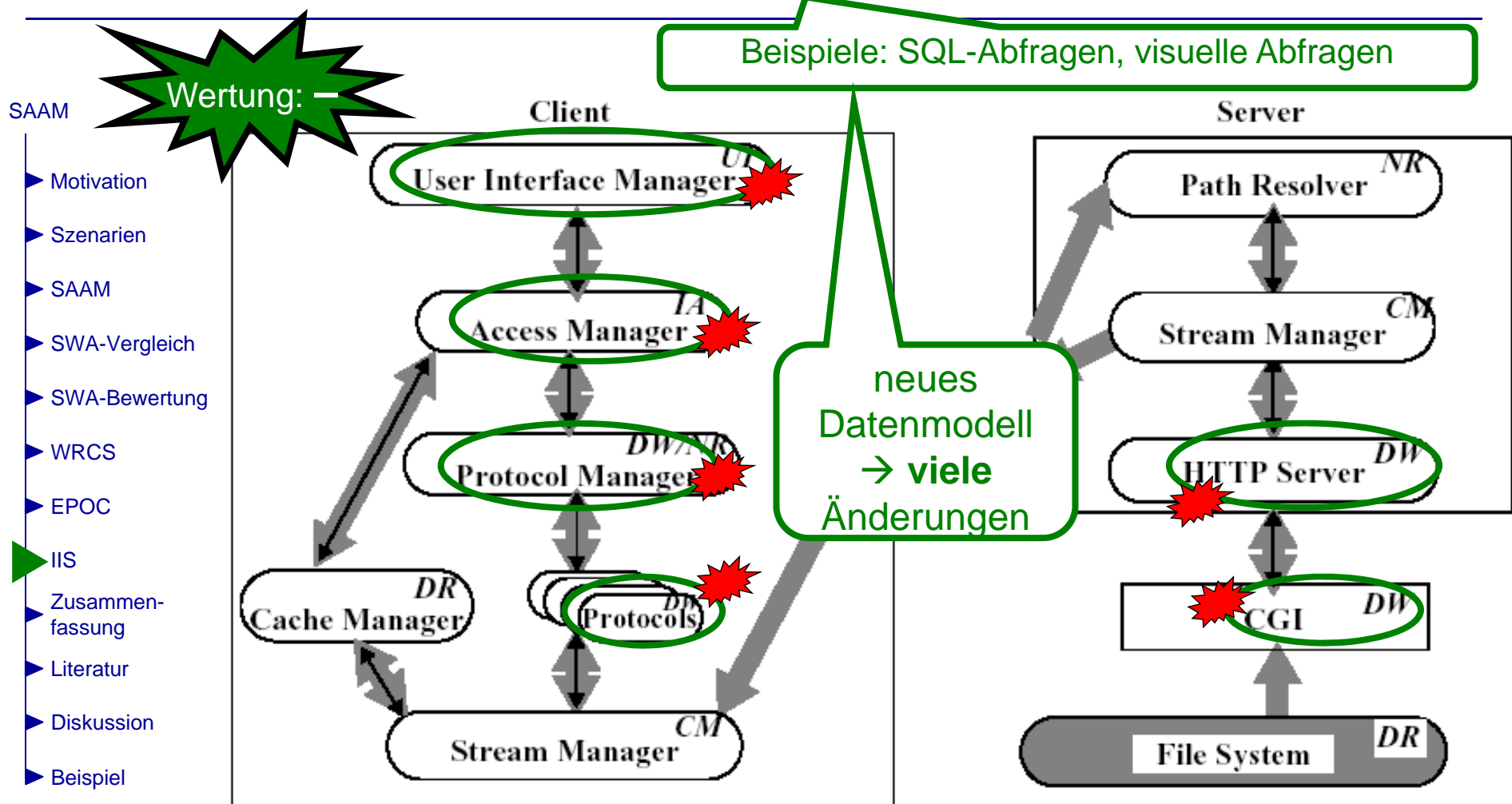


6. WWW: neuer Informationstyp (DW, DR)





6. WWW: neuer Zugriffsmechanismus (UI, IA, DW)

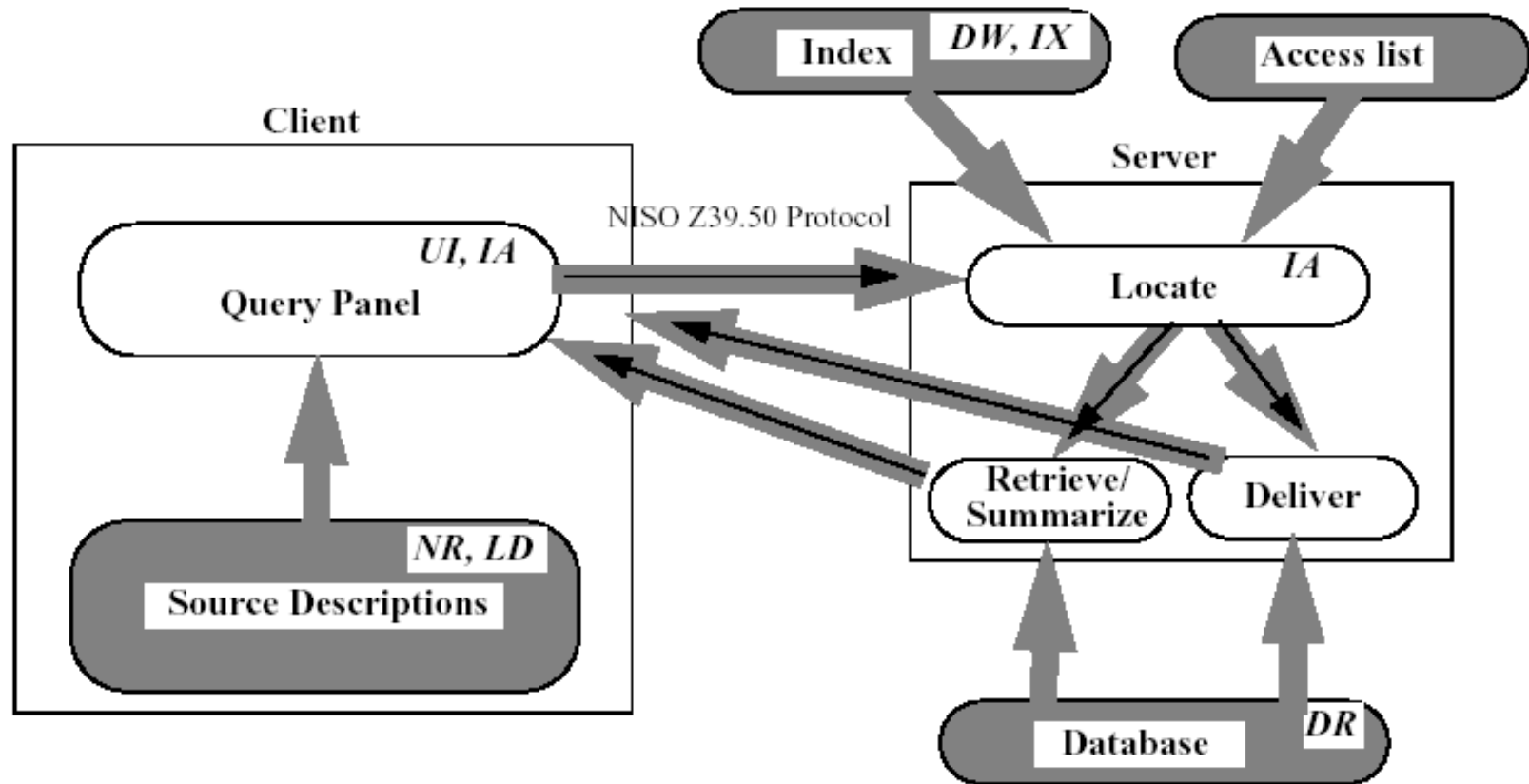




4.+5. WAIS-Architektur (einzelner Server mit einzelner Datenbank)

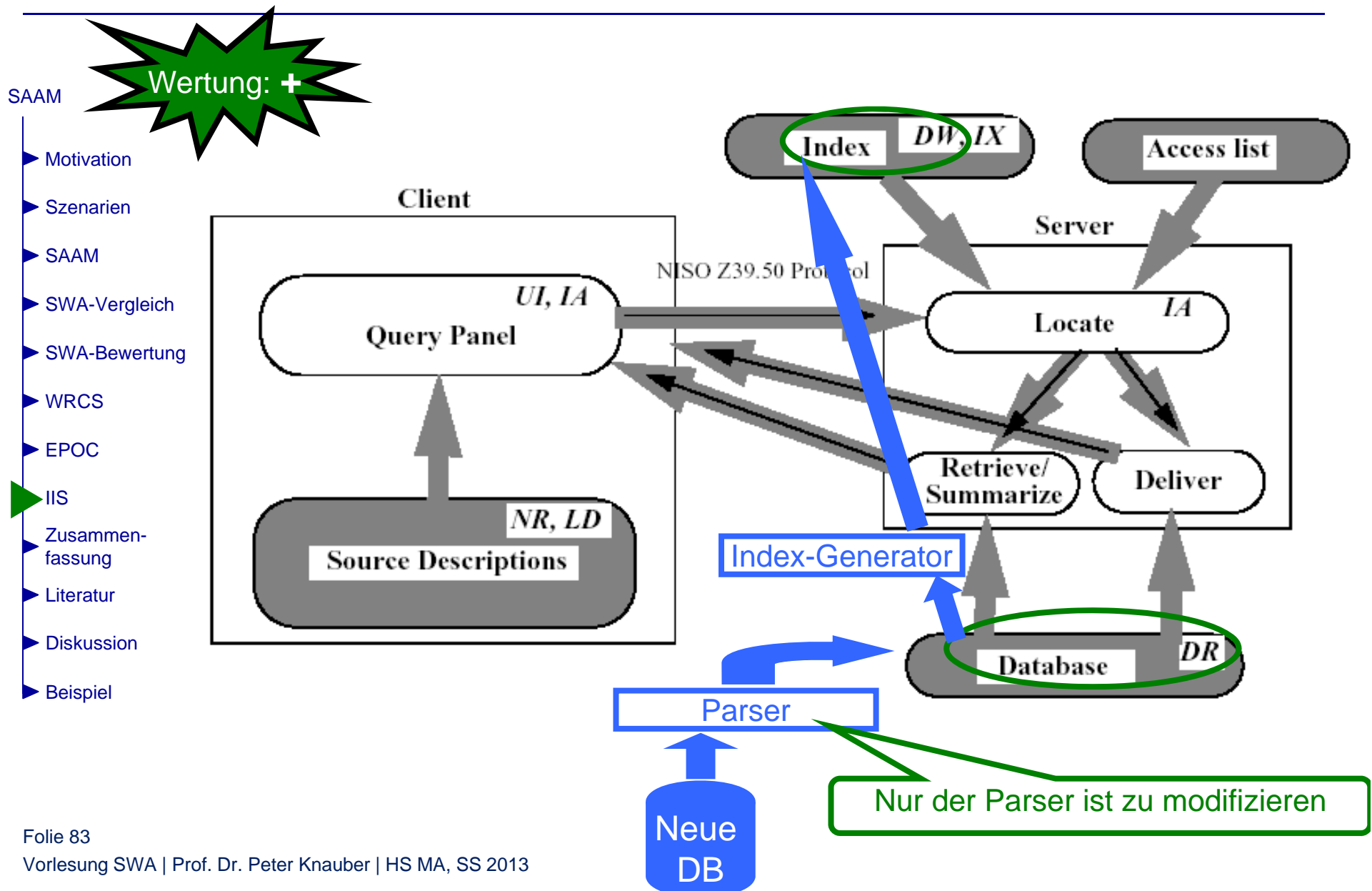
SAAM

- Motivation
- Szenarien
- SAAM
- SWA-Vergleich
- SWA-Bewertung
- WRCS
- EPOC
- IIS
- Zusammenfassung
- Literatur
- Diskussion
- Beispiel



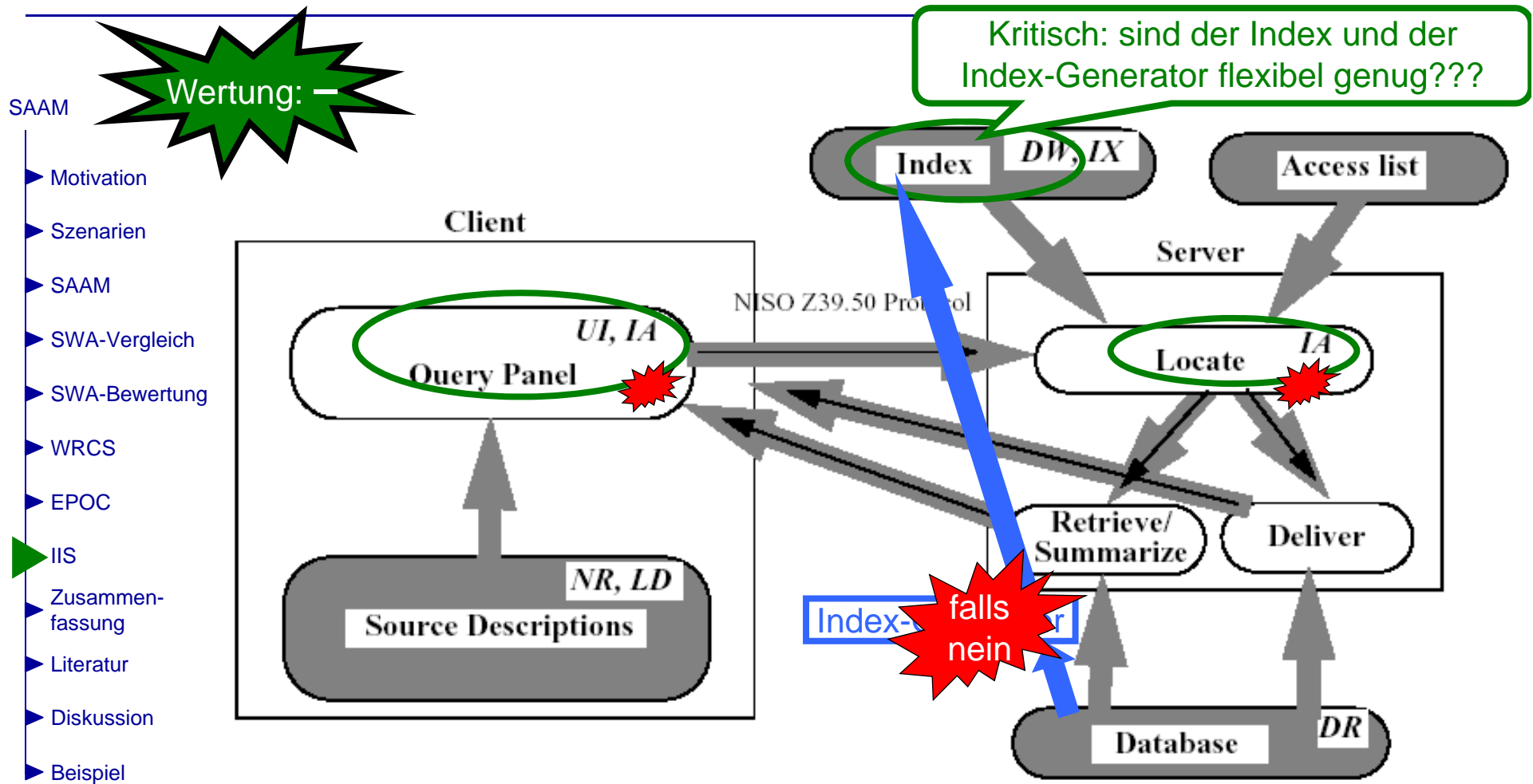


6. WAIS: neuer Informationstyp (DW, DR)





6. WAIS: neuer Zugriffsmechanismus (UI, IA, DW)

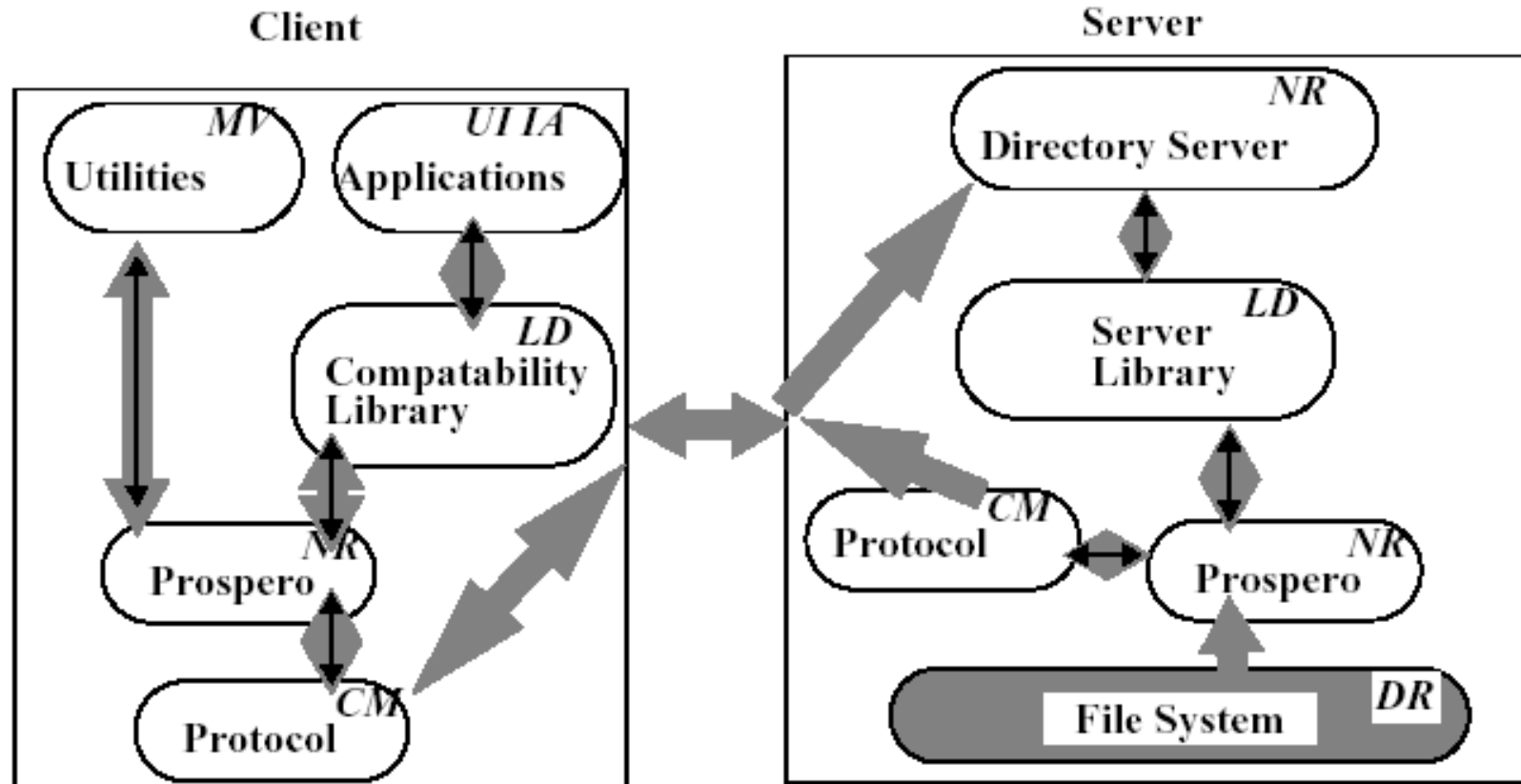




4.+5. Prospero-Architektur

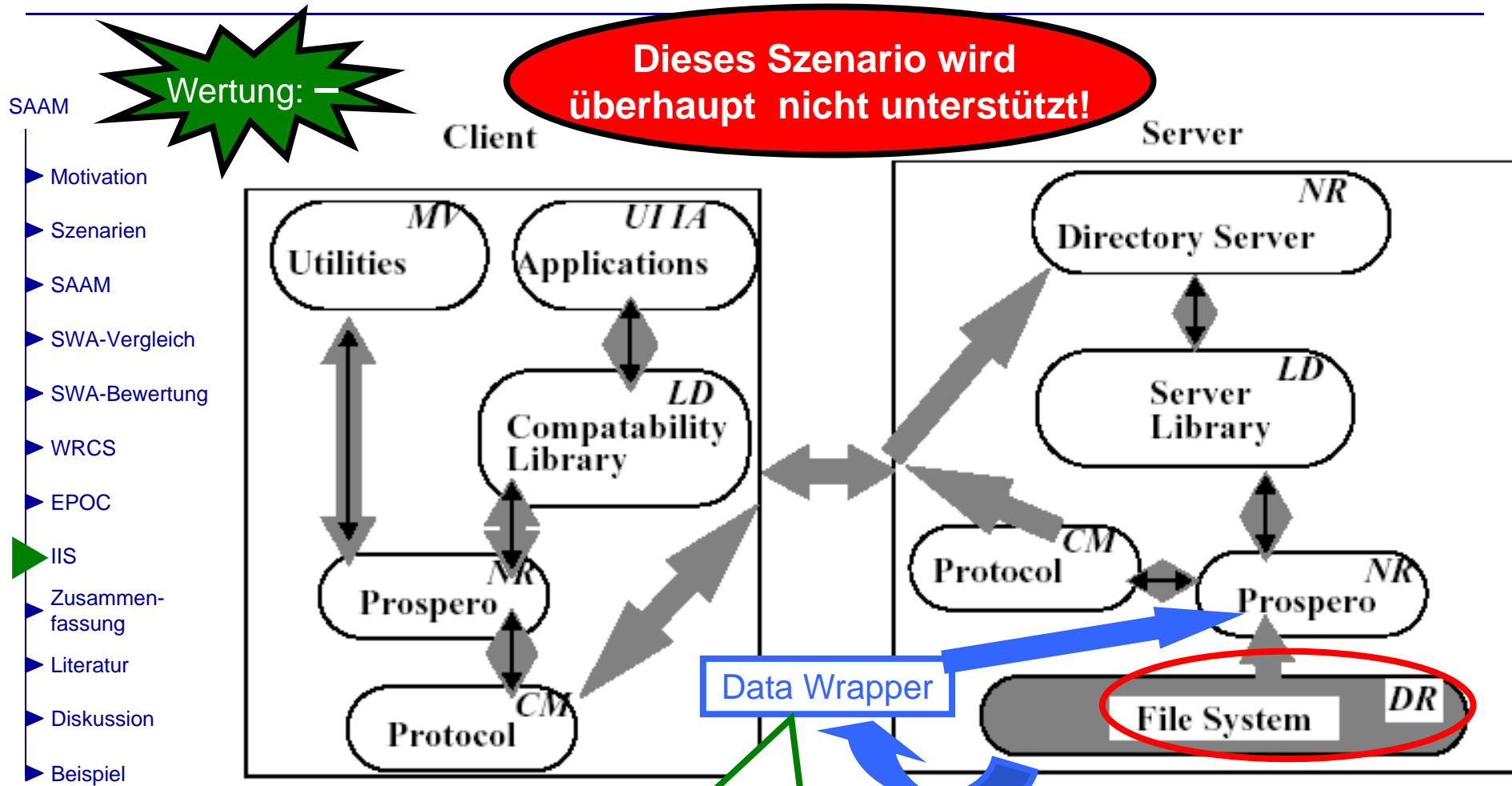
SAAM

- ▶ Motivation
- ▶ Szenarien
- ▶ SAAM
- ▶ SWA-Vergleich
- ▶ SWA-Bewertung
- ▶ WRCS
- ▶ EPOC
- ▶ IIS
- ▶ Zusammenfassung
- ▶ Literatur
- ▶ Diskussion
- ▶ Beispiel





6. Prospero: neuer Informationstyp (DW, DR)



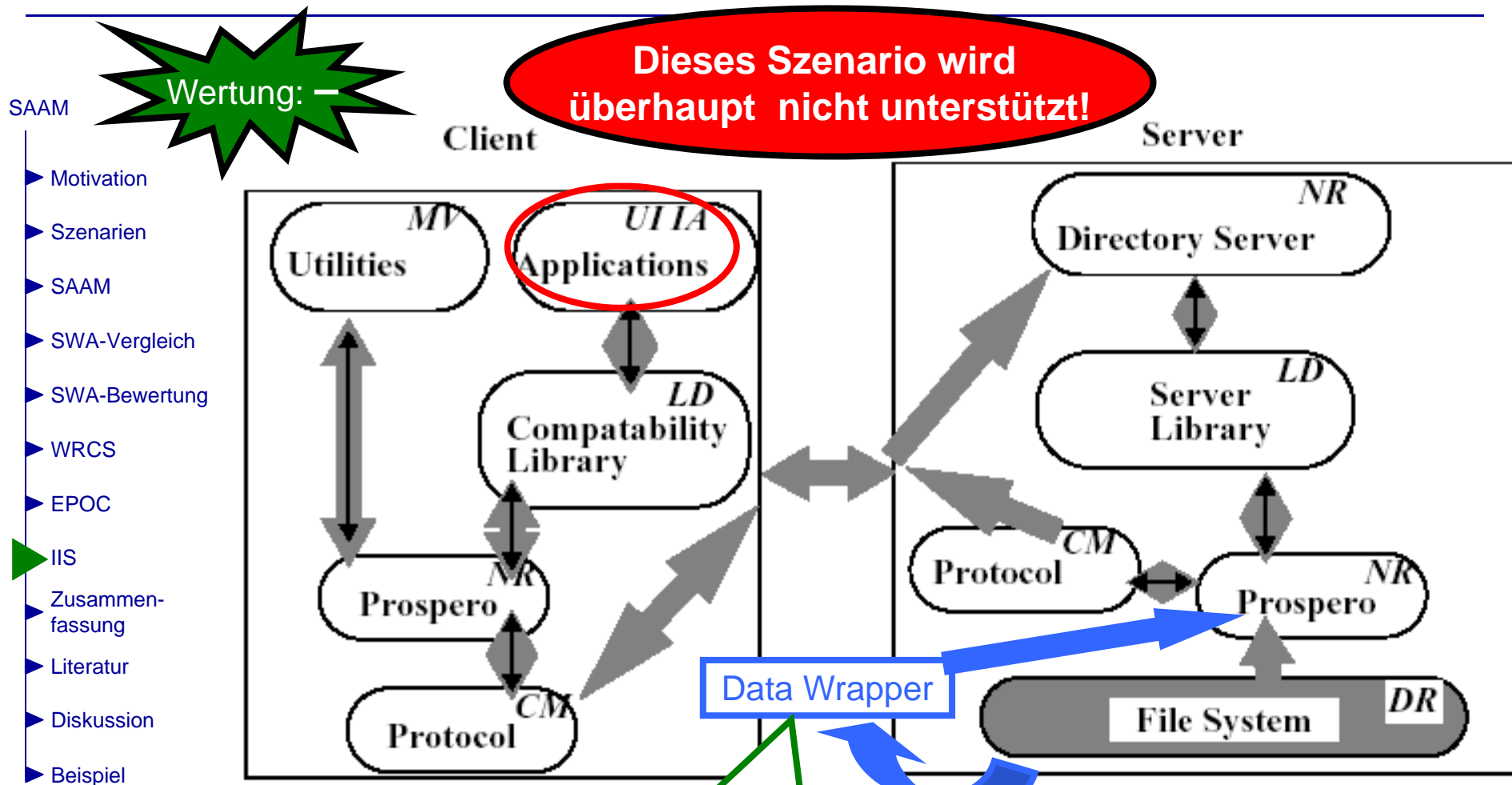
SAAM

- Motivation
- Szenarien
- SAAM
- SWA-Vergleich
- SWA-Bewertung
- WRCS
- EPOC
- IIS
- Zusammenfassung
- Literatur
- Diskussion
- Beispiel

Es existiert kein Data Wrapper (DW)!
Dieser müsste **neu** kreiert und in das System eingebaut werden!



6. Prospero: neuer Zugriffsmechanismus (UI, IA, DW)



SAAM

- Motivation
- Szenarien
- SAAM
- SWA-Vergleich
- SWA-Bewertung
- WRCS
- EPOC
- IIS**
- Zusammenfassung
- Literatur
- Diskussion
- Beispiel

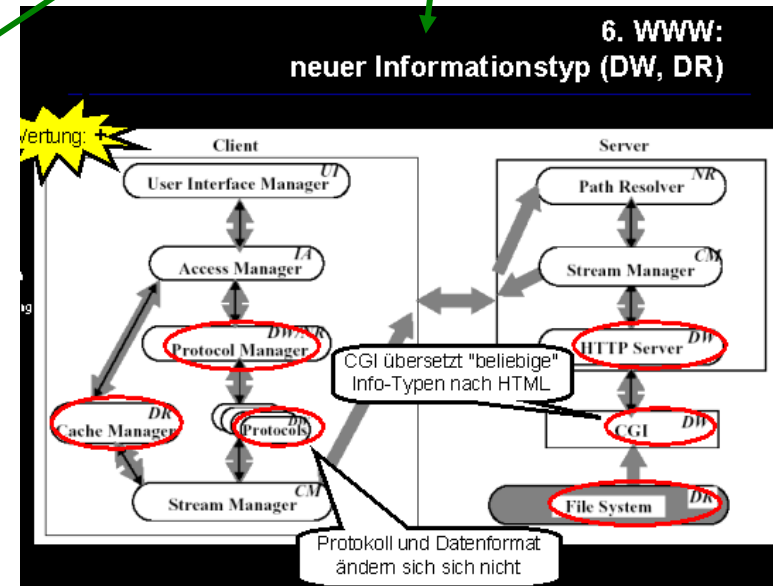
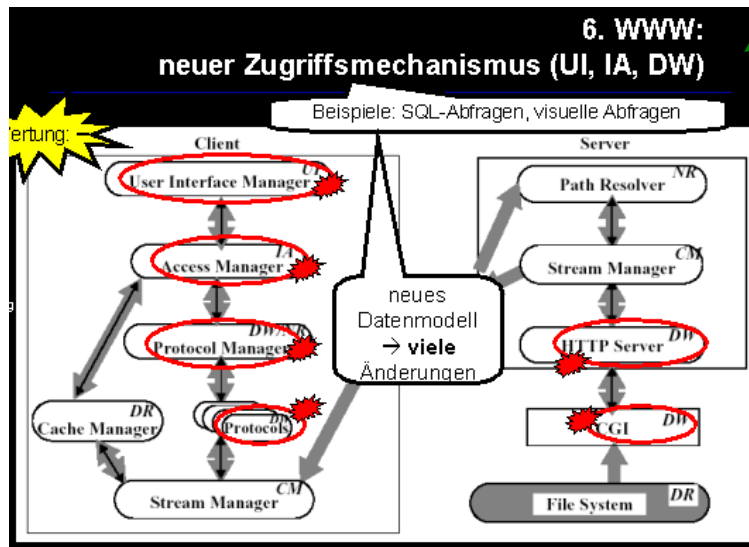


Ergebnis der Analyse der indirekten Szenarien

SAAM

- ▶ Motivation
- ▶ Szenarien
- ▶ SAAM
- ▶ SWA-Vergleich
- ▶ SWA-Bewertung
- ▶ WRCS
- ▶ EPOC
- ▶ IIS
- ▶ Zusammenfassung
- ▶ Literatur
- ▶ Diskussion
- ▶ Beispiel

	Indirekte Szenarien		Szenario-Interaktion	Anzahl von Szenario-Interaktionen
	Szenario h	Szenario i		
WWW	+	-	-	4
WAIS	+	-	0	1
Prospero	-	-	n/a	1 zus. Komponente





Einschränkende Überlegungen

SAAM

- ▶ Motivation
- ▶ Szenarien
- ▶ SAAM
- ▶ SWA-Vergleich
- ▶ SWA-Bewertung
- ▶ WRCS
- ▶ EPOC
- ▶ IIS
- ▶ Zusammenfassung
- ▶ Literatur
- ▶ Diskussion
- ▶ Beispiel

- Eine Architektur *kann nicht* die geforderte Funktionalität oder Qualität *garantieren*
- Qualitätsmerkmale können *nicht ausschließlich* auf Basis einer Architekturbeschreibung gemessen werden
- Architekturbewertung ist ein *notwendiger*, aber *nicht ausreichender* Teil der Bewertung der Gesamtqualität eines Systems

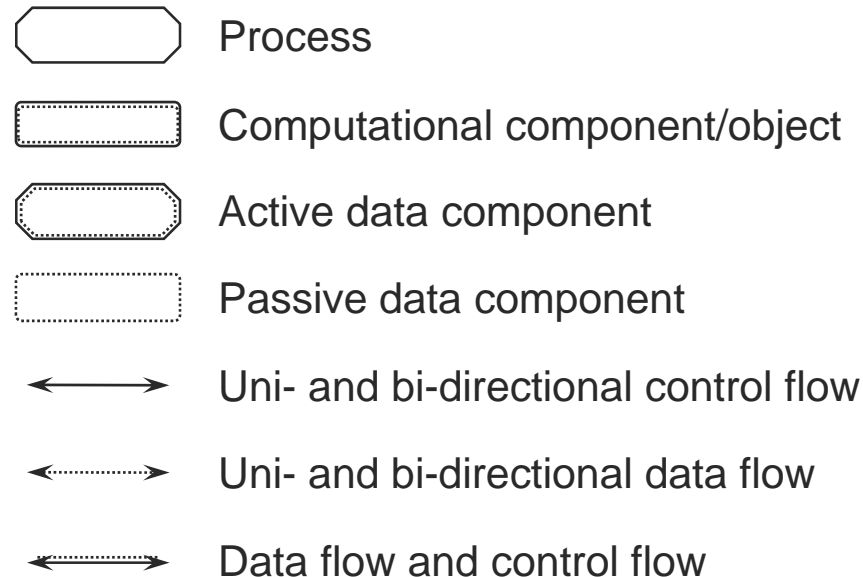


Probleme bei einer Architekturbewertung und ihre Lösung mittels SAAM 1/3

SAAM

- ▶ Motivation
- ▶ Szenarien
- ▶ SAAM
- ▶ SWA-Vergleich
- ▶ SWA-Bewertung
- ▶ WRCS
- ▶ EPOC
- ▶ IIS
- ▶ Zusammenfassung
- ▶ Literatur
- ▶ Diskussion
- ▶ Beispiel

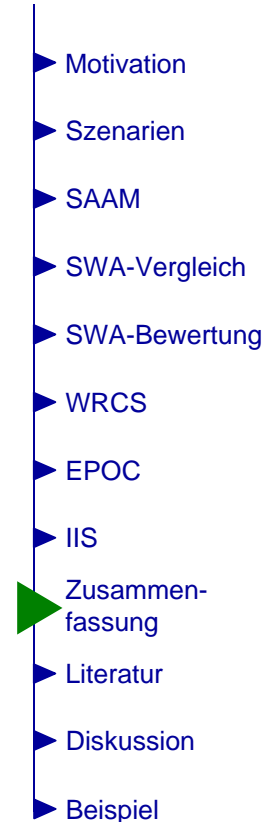
- Verschiedene Architekturen (verschiedener Systeme) benutzen unterschiedliche Beschreibungsformen
- SAAM benutzt eine einheitliche Darstellung für die Beschreibung aller Systeme
- Die gleichartige Beschreibung erlaubt den direkten Vergleich alternativer Architekturen





Probleme bei einer Architekturbewertung und ihre Lösung mittels SAAM 2/3

SAAM



- Qualitätsattribute sind zu abstrakt, um sie direkt messen zu können
 - SAAM benutzt Szenarien, um Qualitätsattribute zu konkretisieren
 - Jedes Szenario beschreibt eine Schrittfolge für die Nutzung oder Modifikation eines Systems
 - Jedes Szenario repräsentiert eine Klasse ähnlicher Szenarien
 - Jedes Szenario ist konkret genug für eine sinnvolle Analyse

Für jedes Szenario gibt es eine relative Reihenfolge der Architekturalternativen



Probleme bei einer Architekturbewertung und ihre Lösung mittels SAAM 3/3

SAAM

- ▶ Motivation
- ▶ Szenarien
- ▶ SAAM
- ▶ SWA-Vergleich
- ▶ SWA-Bewertung
- ▶ WRCS
- ▶ EPOC
- ▶ IIS
- ▶ Zusammenfassung
- ▶ Literatur
- ▶ Diskussion
- ▶ Beispiel

- Schwächen der Architektur werden normalerweise erst entdeckt, wenn sie direkte, negative Auswirkungen zeigen

→ SAAM erlaubt die Analyse vor der Implementierung

→ SAAM ist unabhängig von einer Entwicklungsmethode, einer Programmiersprache und von Implementierungsdetails

→ Alles was SAAM benötigt ist eine Beschreibung, die ausreicht, um die Szenarien anzuwenden, *aber*

→ Zusätzliche Informationen können genutzt werden, um SAAM-Ergebnisse genauer zu machen



Grenzen von SAAM

SAAM

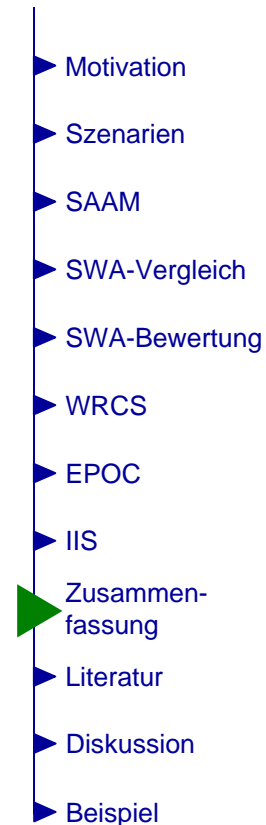
- ▶ Motivation
- ▶ Szenarien
- ▶ SAAM
- ▶ SWA-Vergleich
- ▶ SWA-Bewertung
- ▶ WRCS
- ▶ EPOC
- ▶ IIS
- ▶ Zusammenfassung
- ▶ Literatur
- ▶ Diskussion
- ▶ Beispiel

- Architektur-unabhängige Potenziale werden nicht berücksichtigt; Beispiele
 - Domain-spezifische Sprachen
 - Generatoren, Anwendungsgeneratoren und andere Tools
- Die Architekturen der Kandidatensysteme können ungeschickt dargestellt sein
- Die Vollständigkeit der Szenarien kann nicht geprüft werden
- Nicht alle Qualitätsattribute können (alleine) an der Architektur gemessen werden; Beispiele
 - Performance
 - Bedienbarkeit
- **Eine gute Architektur (alleine) kann die Qualität eines Systems nicht garantieren!**



Vorteile von SAAM

SAAM



- SAAM erzwingt (über die Stakeholder) das Einbeziehen des organisatorischen Kontexts
- Abstrakte Qualitätsattribute werden durch Szenarien auf eine konkrete Art repräsentiert
- SAAM kann sehr früh im Entwicklungsprozess angewandt werden



Erfahrungen mit SAAM

SAAM

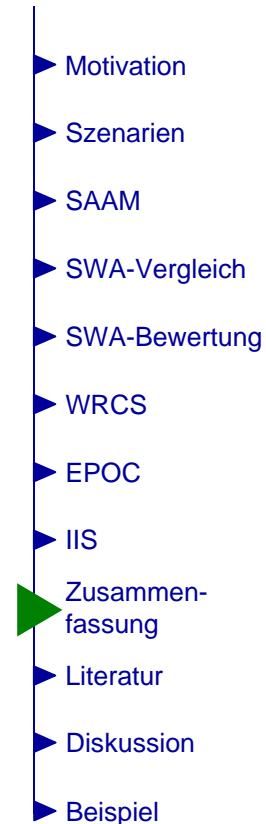
- ▶ Motivation
- ▶ Szenarien
- ▶ SAAM
- ▶ SWA-Vergleich
- ▶ SWA-Bewertung
- ▶ WRCS
- ▶ EPOC
- ▶ IIS
- ▶ Zusammenfassung
- ▶ Literatur
- ▶ Diskussion
- ▶ Beispiel

- Direkte Szenarien liefern
 - Sehr gute Unterscheidungskriterien für die zu vergleichenden Architekturen
 - Einen Mechanismus für die Erfassung und das Verständnis der (statischen und dynamischen) Sichten auf die Architekturen
- Es ist wichtig, Stakeholder bei den Evaluierungstreffen dabei zu haben
 - Die Moderatoren (Evaluatoren) haben nicht die domänen-spezifische Erfahrung, um die Architekten kritisch zu befragen
 - Stakeholder finden es lehrreich
- SAAM im Vergleich zu traditionellen Architektur-Metriken
 - Viele Szenario-Interaktionen zeigen niedrige Kohäsion
 - Ein Szenario mit breitgestreuten Auswirkungen zeigt hohe Kopplung
 - Kopplungs- und Kohäsionsmetriken korrespondieren nicht mit verschiedenen *Nutzungen*
 - SAAM macht solche Metriken spezifischer



Zusammenfassung: Eine SAAM-Evaluierung liefert

SAAM



- Technische Ergebnisse:
Einsichten in die Fähigkeiten und Grenzen der Systeme
- Soziale Ergebnisse:
 - Architektur-Dokumentation wird erzwungen
 - Kommunikationsmittel / -Medium zwischen den Stakeholders



SAAM

- ▶ Motivation
- ▶ Szenarien
- ▶ SAAM
- ▶ SWA-Vergleich
- ▶ SWA-Bewertung
- ▶ WRCS
- ▶ EPOC
- ▶ IIS
- ▶ Zusammenfassung
- ▶ Literatur
- ▶ Diskussion
- ▶ Beispiel

- G. Abowd, L. Bass, P. Clements, R. Kazman: MORALE METHODOLOGY GUIDEBOOK Software Architecture Analysis Method, <http://www.cc.gatech.edu/morale/> (geprüft im Mai '05)
- P. Clements, L. Bass, R. Kazman, G. Abowd: Predicting Software Quality by Architecture-Level Evaluation, Proceedings of the Fifth International Conference on Software Quality, 1995
- R. Kazman, G. Abowd, L. Bass, P. Clements: Scenario-Based Analysis of Software Architecture, Technical Report CS-95-45, Department of Computer Science, University of Waterloo, 1995
- R. Kazman, G. Abowd, L. Bass, P. Clements: Scenario-Based Analysis of Software Architecture, IEEE Software, 11/1996



SAAM

- ▶ Motivation
- ▶ Szenarien
- ▶ SAAM
- ▶ SWA-Vergleich
- ▶ SWA-Bewertung
- ▶ WRCS
- ▶ EPOC
- ▶ IIS
- ▶ Zusammenfassung
- ▶ **Literatur**
- ▶ Diskussion
- ▶ Beispiel

- **[KBAC 95]** R. Kazman, L. Bass, G. Abowd, P. Clements: An Architectural Analysis Case Study: Internet information Systems, First International Workshop on Architectures for Software Systems, 1995
- R. Kazman, L. Bass, G. Abowd, M. Webb: SAAM: A Method for Analyzing the Properties of Software Architectures, Proceedings of the 16th International Conference on Software Engineering, 1994
- **[DeSK 95]** M. De Simone, R. Kazman: Software Architecture Analysis: An Experience Report, Proceedings of CASCON'95, Toronto, 1995
- **[Talgren 95]** Tapio Talgren: EPOC Assessment Report, ESAPS-Projekt WP 1, Task 1.1, 2001, <http://www.esi.es/en/Projects/esaps/publicResults.html> (geprüft im Mai '05)