

Architecture Influence Cycle (AIC)

Siklus Pengaruh Arsitektur

ER234403 – Arsitektur Perangkat Lunak

ER234403 – *Software Architecture*

Outline

- Latar Belakang AIC
- Faktor yang Mempengaruhi Arsitektur
- Faktor yang Dipengaruhi Arsitektur
- Siklus Pengaruh (Cycle of Influences)
- Studi Kasus
- Ringkasan

Latar Belakang AIC

- Software architecture **tidak berdiri sendiri**.
- Arsitektur **dipengaruhi oleh banyak faktor**: stakeholder, tujuan bisnis, lingkungan teknis, pengalaman arsitek.
- Arsitektur juga **mempengaruhi kembali** faktor-faktor tersebut.

Apa itu Architecture Influence Cycle?

- *Architecture Influence Cycle (AIC)* = **siklus sebab-akibat** antara faktor yang membentuk arsitektur dan dampak arsitektur pada organisasi & sistem.
- Menunjukkan bahwa arsitektur adalah bagian dari **ekosistem yang dinamis**, bukan keputusan sekali jalan.

Mengapa Harus Dipahami?

- Membantu arsitek **memahami konteks** sebelum mengambil keputusan desain.
- Memahami adanya **trade-off**: keputusan teknis dipengaruhi strategi bisnis, dan sebaliknya.
- Menjadi dasar untuk **evaluasi arsitektur** dalam jangka panjang.

Faktor yang Mempengaruhi Arsitektur

Arsitektur perangkat lunak **dibentuk oleh banyak faktor:**

1. **Stakeholders**
2. **Business / Mission Goals**
3. **Technical Environment**
4. **Architect's Experience**

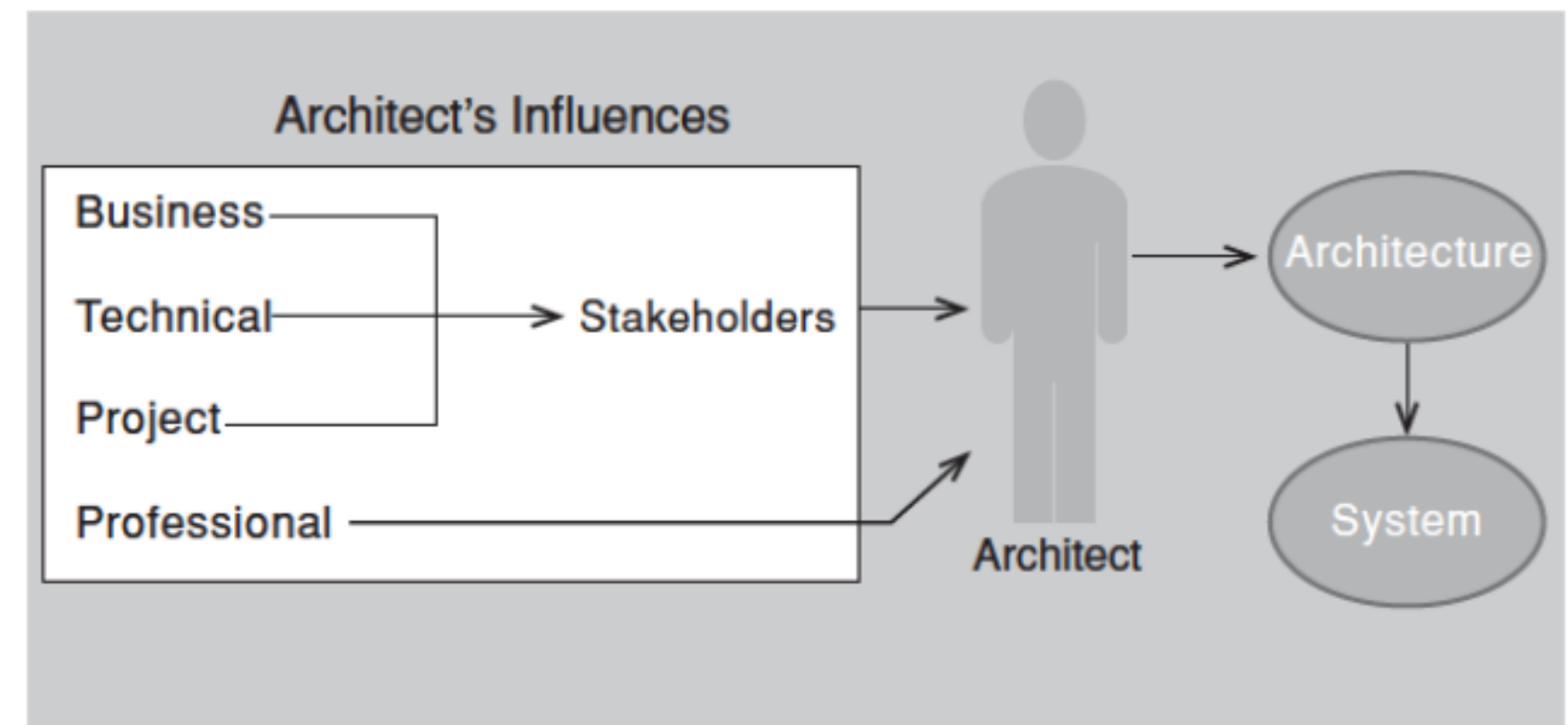


FIGURE 3.4 Influences on the architect

All citations are from the course textbook, *Software Architecture in Practice*, Third Edition, unless otherwise noted.

Stakeholders

- Stakeholder/pemangku kepentingan adalah orang-orang yang memiliki kepentingan pada pengembangan sistem software.
- Termasuk:
 - pelanggan/customer,
 - pengguna akhir/end-user
 - manajer,
 - developer,
 - maintainer,
 - marketer.
- Masing-masing membawa **kebutuhan & prioritas berbeda** yang ingin dioptimalkan.

Pelanggan (Customers)

- Pelanggan adalah orang/organisasi yang membayar pengembangan sistem
- Fokus utama:
 - Biaya, fungsionalitas, dan manfaat dari sistem.
 - Development time/time to market
 - Kualitas dan fleksibilitas
- Ingin arsitektur yang **efisien** dan memberikan **nilai bisnis**.
- Pelanggan tidak selalu sama dengan end-users/pengguna.
- Pertanyaan umum:
 - “Berapa biaya sistem ini?”
 - “Apakah arsitektur ini mendukung ROI yang baik?”

Pengguna (Users)

- **Pengguna** adalah orang yang **berinteraksi langsung dengan sistem**.
- **Fokus utama:**
 - Fungsi yang relevan dengan kebutuhan mereka.
 - Kemudahan penggunaan (*usability*).
 - Respons cepat & reliabilitas.
- Ingin arsitektur yang menghasilkan **sistem handal** dan **mudah digunakan**.
- Pengguna sering berbeda dengan pelanggan (tidak selalu orang yang membayar).
- **Pertanyaan umum:**
 - “Apakah sistem ini mudah dipelajari dan digunakan?”
 - “Apakah sistem cepat merespons permintaan saya?”

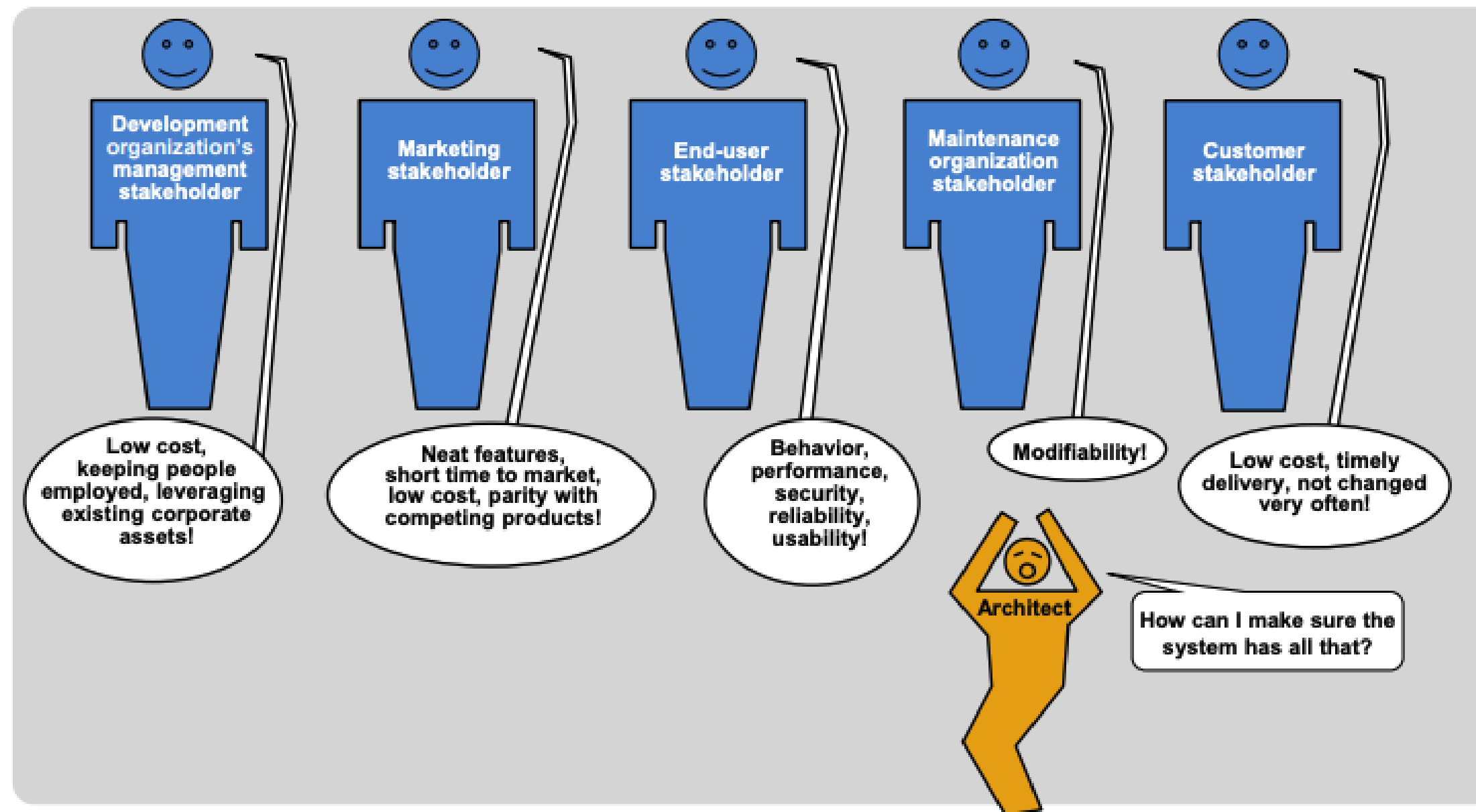
Manajemen (Management)

- **Manajemen** adalah pihak yang bertanggung jawab pada **strategi bisnis dan pengambilan keputusan organisasi**.
- **Fokus utama:**
 - Time-to-market (seberapa cepat sistem diluncurkan).
 - Profitabilitas & efisiensi biaya.
 - Mitigasi risiko teknis & bisnis.
- Ingin arsitektur yang memungkinkan **kontrol biaya, risiko rendah, dan keuntungan kompetitif**.
- **Pertanyaan umum:**
 - “Berapa lama produk ini bisa diluncurkan ke pasar?”
 - “Apa risiko terbesar dari desain ini?”

Stakeholder Lainnya

- **Developer:** menginginkan arsitektur yang konsisten, terdokumentasi, mudah dipahami.
- **Maintainer:** menginginkan sistem yang mudah diperbaiki & dikembangkan.
- **Tester:** membutuhkan sistem yang dapat diuji dengan mudah.
- **Fokus utama:**
 - Produktivitas tim pengembang.
 - Kemudahan pemeliharaan jangka panjang.
 - Dukungan operasional yang lancar.
- **Pertanyaan umum:**
 - “Apakah arsitektur ini mudah dikembangkan & diuji?”
 - “Apakah sistem ini mudah dipelihara dan dioperasikan?”

Kepentingan Stakeholders



Business / Mission Goals

- Arsitektur dipengaruhi oleh strategi & tujuan bisnis organisasi.
- Faktor penting:
 - **Time-to-market** → kecepatan rilis produk.
 - **Cost constraints** → keterbatasan anggaran.
 - **Strategic goals** → keberlanjutan, diferensiasi, inovasi.
- Contoh: Startup memilih arsitektur sederhana agar software cepat diluncurkan ke pengguna.

Faktor-faktor Bisnis

- Time to market
- Rollout schedule
- Use of legacy systems
- Available expertise
- Support for existing products
- Targeted markets
- Political interests
- Existing architectures
- Plans for long-term infrastructure
- Organizational structure
- Projected lifetime of the system
- Workforce utilization
- Cost
- Investment in existing assets

Technical Environment

- Teknologi yang tersedia & digunakan dalam organisasi.
- Termasuk: bahasa pemrograman, framework, middleware, infrastruktur, tools.
- Contoh:
 - Organisasi berbasis cloud: cenderung memilih microservices.
 - Organisasi dengan legacy system: sering terbatas pilihan teknologinya.

Contoh Kasus Technical Environment

Arsitektur perangkat lunak dipengaruhi oleh kondisi teknis yang tersedia dalam organisasi, seperti:

1. Bahasa pemrograman & framework

- Contoh: Java Spring Boot, .NET Core, Python Django

2. Middleware & platform

- Contoh: Message broker (Kafka, RabbitMQ), container (Docker, Kubernetes)

3. Infrastruktur & deployment target

- Contoh: Cloud provider (AWS, GCP, Azure) vs on-premises

4. Integrasi dengan sistem legacy

- Contoh: Mainframe, ERP lama, database lama (Oracle, DB2)

5. Tools & praktik pengembangan

- Contoh: CI/CD pipeline, DevOps, automated testing

Contoh kasus:

Organisasi yang sudah berinvestasi besar di **Microsoft stack** (Windows Server, SQL Server, .NET) biasanya akan memilih arsitektur berbasis **.NET ecosystem** daripada membangun sistem baru dengan Java.

Architect's Experience

- Pengalaman arsitek sangat berpengaruh pada desain:
 - **Proyek sukses/gagal** sebelumnya.
 - **Pengetahuan teknologi** terbaru.
 - **Best practices** yang pernah dipakai.
 - **Latar belakang Pendidikan/pelatihan** yang pernah dijalani.
- Contoh: Arsitek yang pernah gagal dengan monolith cenderung mendorong microservices.

Faktor yang Dipengaruhi Arsitektur

- **Stakeholder requirements:** arsitektur dapat mengubah ekspektasi kebutuhan
- **Development organization environment:** arsitektur menentukan struktur tim, aktivitas integrasi, alokasi sumber daya
- **Architect's experience:** proyek sukses/gagal memengaruhi desain di masa depan
- **Technical environment:** arsitektur & teknologi baru bisa membentuk fondasi lingkungan teknis (contoh: relational DB, WWW, microservices, container)

Stakeholder Requirements

- Arsitektur perangkat lunak dipengaruhi oleh kebutuhan stakeholder, namun arsitektur juga dapat **membentuk ulang ekspektasi** stakeholder.
- **Studi Kasus:**
 - Stakeholder datang dengan kebutuhan awal (fitur, kualitas).
 - Arsitektur bisa memperlihatkan **kemungkinan & batasan teknis**.
 - Hasilnya, kebutuhan bisa berubah/beradaptasi.
- **Contoh:**
 - User awalnya minta **respon < 1 detik** untuk semua fitur.
 - Arsitektur sistem berbasis microservices menunjukkan hal itu mahal & kompleks. Kebutuhan diubah menjadi **respon cepat hanya untuk fitur utama**.

Development Organization Environment

- Arsitektur memengaruhi **cara organisasi pengembang bekerja**.
- Struktur arsitektur memengaruhi struktur tim (*Conway's Law*).
- Arsitektur menentukan **alur kerja, integrasi, dan pembagian tanggung jawab**.
- **Contoh:**
 - Sistem berbasis **microservices**: tim dipecah menjadi **tim kecil independen**.
 - Sistem **monolith**: satu tim besar bekerja pada codebase tunggal.

Architect's Experience

- Pengalaman arsitek, baik sukses maupun gagal, sangat memengaruhi desain berikutnya.
- Arsitek menggunakan **heuristik & pola** dari pengalaman sebelumnya.
- Keputusan arsitektur sering kali dipengaruhi oleh **preferensi pribadi & track record**.
- **Contoh:**
 - Arsitek yang pernah gagal dengan **monolith** pada sistem skala besar akan lebih cenderung memilih **microservices**.
 - Arsitek berpengalaman di **event-driven systems** akan lebih memilih menggunakan message bus.

Technical Environment

- Arsitektur dipengaruhi oleh teknologi yang tersedia & juga **membentuk lingkungan teknis baru**.
- Keputusan arsitektur memunculkan standar teknis baru.
- Teknologi yang tersedia dapat membatasi atau memperluas opsi desain.
- **Contoh:**
 - Kemunculan **relational database** mendorong arsitektur berbasis data.
 - **WWW & REST APIs** melahirkan arsitektur web modern.
 - **Cloud-native & container orchestration** mempercepat adopsi **microservices**.

Siklus Pengaruh (Cycle of Influences)

- Hubungan saling memengaruhi antara:
 - Business/mission goals
 - Stakeholder requirements
 - Architect's experience
 - Architectures
 - Fielded systems
- Adanya **feedback loop** yang bisa dimanfaatkan organisasi untuk:
 - Mendorong pertumbuhan
 - Memperluas area bisnis/enterprise
 - Mengoptimalkan investasi dari arsitektur sebelumnya

Architecture Influence Cycle (AIC)

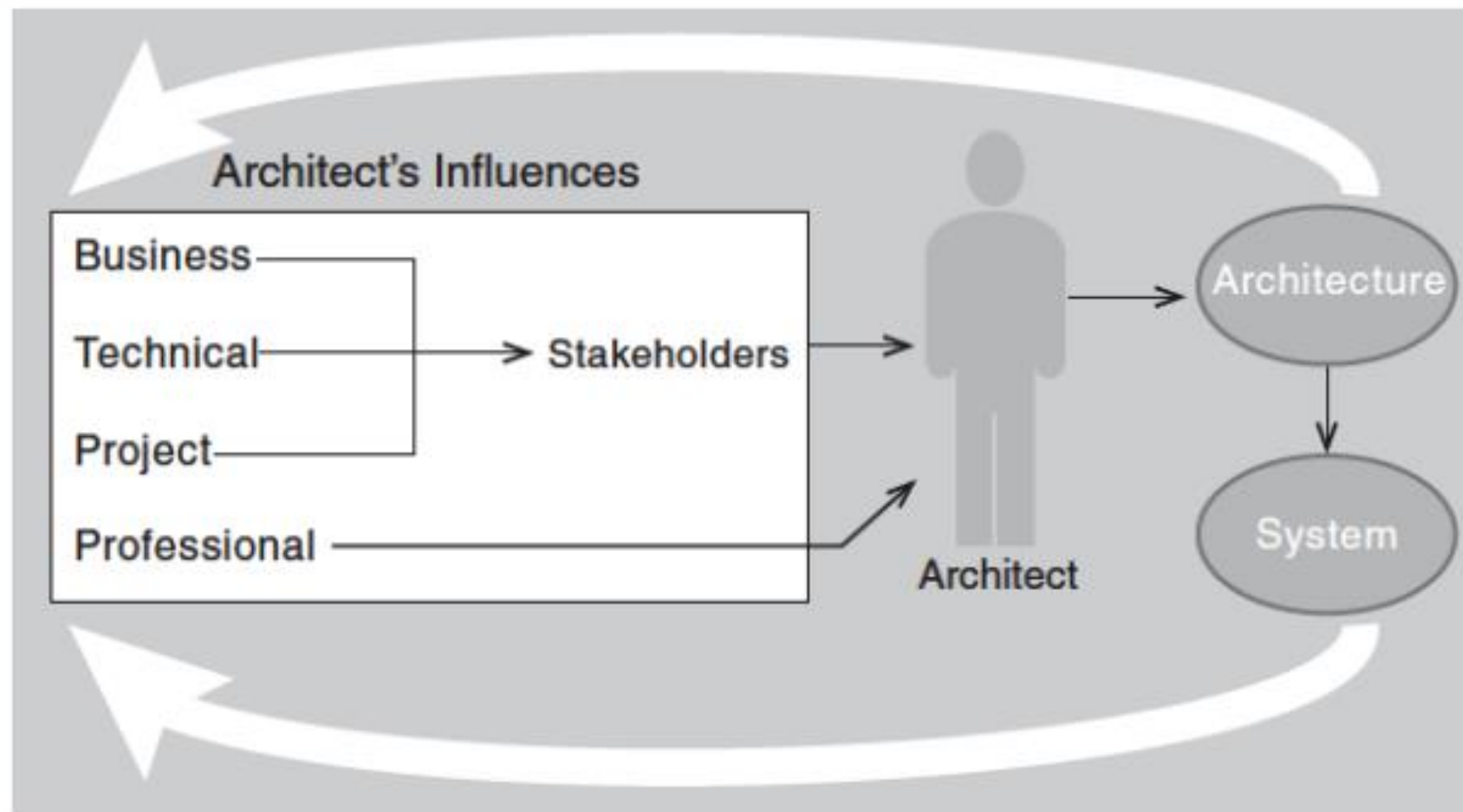


FIGURE 3.5 Architecture Influence Cycle

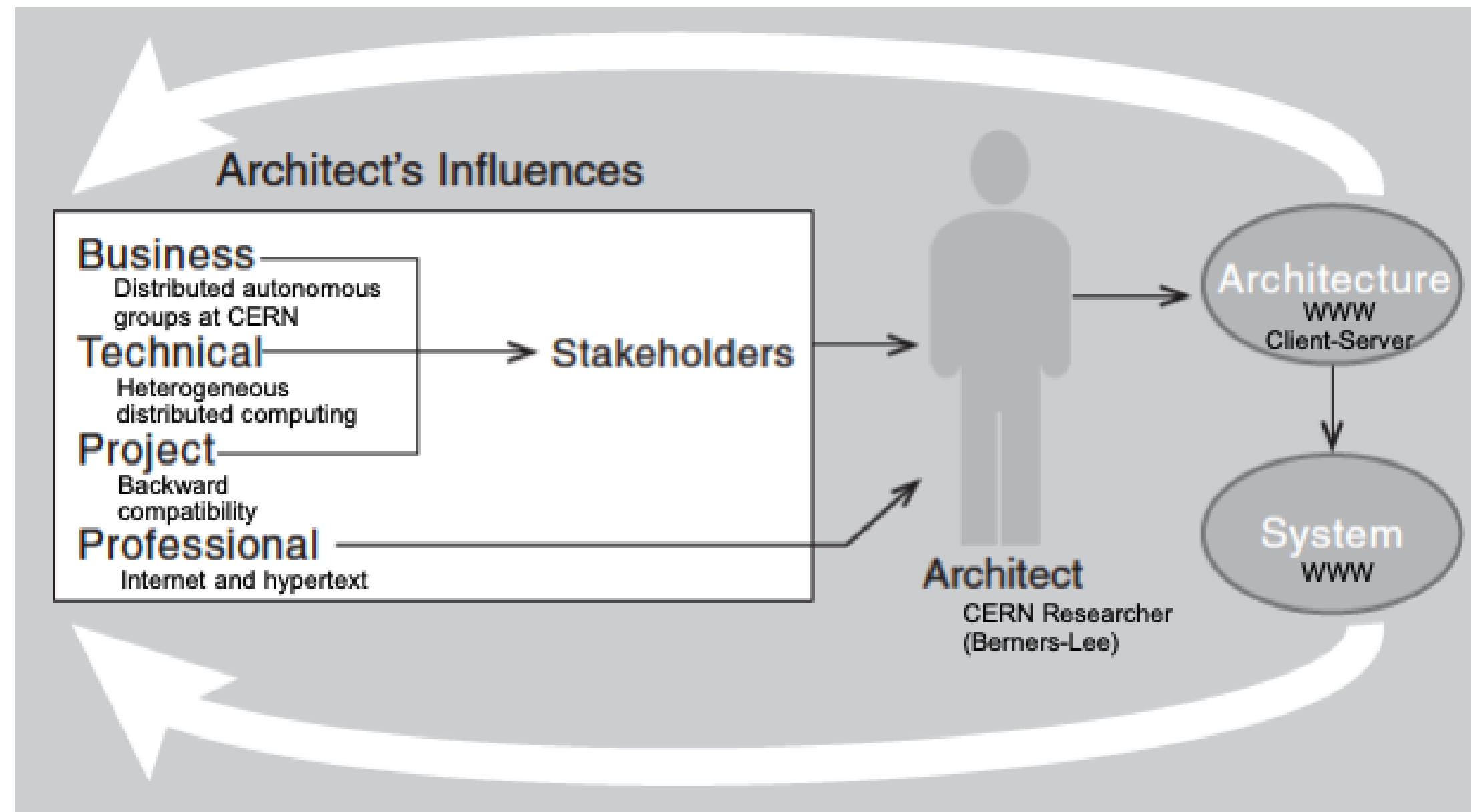
Studi Kasus: World Wide Web (WWW)

- **Latar Belakang**
 - Awalnya dikembangkan untuk berbagi dokumen riset secara mudah.
 - Tujuan: komunikasi & kolaborasi antar peneliti.
- **Faktor yang Mempengaruhi Arsitektur:**
 - **Stakeholders:** peneliti ingin akses mudah & global.
 - **Business/Mission Goals:** open access, interoperabilitas.
 - **Technical Environment:** jaringan internet, protokol TCP/IP.
 - **Architect's Experience:** Tim Berners-Lee berpengalaman dengan hypertext systems.

Dampak Arsitektur pada Faktor Lain

- **Stakeholder requirements:** ekspektasi pengguna berkembang dari sekadar berbagi teks: multimedia, interaktif.
- **Development organization:** melahirkan W3C untuk mengelola standar web.
- **Technical environment:** mendorong terciptanya HTML, HTTP, URL → fondasi ekosistem internet modern.
- **Architect's experience:** pengalaman awal membentuk desain web selanjutnya (REST, linked data).

Initial AIC for the Web



Lessons Learned

- Arsitektur awal yang sederhana **berkembang** menjadi ekosistem global.
- Keputusan desain (misalnya HTTP yang stateless) memengaruhi arah perkembangan internet hingga sekarang.
- Architecture Influence Cycle menunjukkan bahwa **keputusan arsitektur tidak hanya dipengaruhi oleh konteks saat itu, tapi juga membentuk masa depan teknologi.**

Ringkasan

- Arsitektur perangkat lunak **dipengaruhi oleh banyak faktor**:
 - **Stakeholders**: kebutuhan & prioritas (customer, user, manajemen, developer, dsb.)
 - **Business / Mission Goals**: biaya, time-to-market, strategi jangka panjang
 - **Technical Environment**: teknologi, infrastruktur, tools yang tersedia
 - **Architect's Experience**: pengalaman sukses/gagal, preferensi, best practices
- Arsitektur juga **mempengaruhi kembali** faktor-faktor tersebut:
 - Mengubah ekspektasi stakeholder
 - Membentuk organisasi pengembang
 - Memengaruhi ekosistem teknis & teknologi baru
- **Architecture Influence Cycle** = hubungan timbal balik yang terus berulang.