Cloud Computing - FinOps

Wojciech Barczyński

Sprawdzenie obecności

Slajdy

→ github.com/wojciech11/se_cloud_finops

O mnie

- VP of Engineering Platform @ Kubermatic
- Poprzednio: VP of Engineering @ Spacelift (Series C)
- Doświaczenie startupy i scale-ups, AI/ML, B2B i B2C
- 1100+ h szkoleń, 30+ wystąpień na konferencjach
- *„Wszystko jest iteracją, wszystko jest eksperymentem"*

Agenda

1. Co to jest FinOps?
2. Usługi chmurowe
3. Koszty z punktu widzenia technicznego
4. Koszty vs innowacja
5. FinOps Framework

FinOps

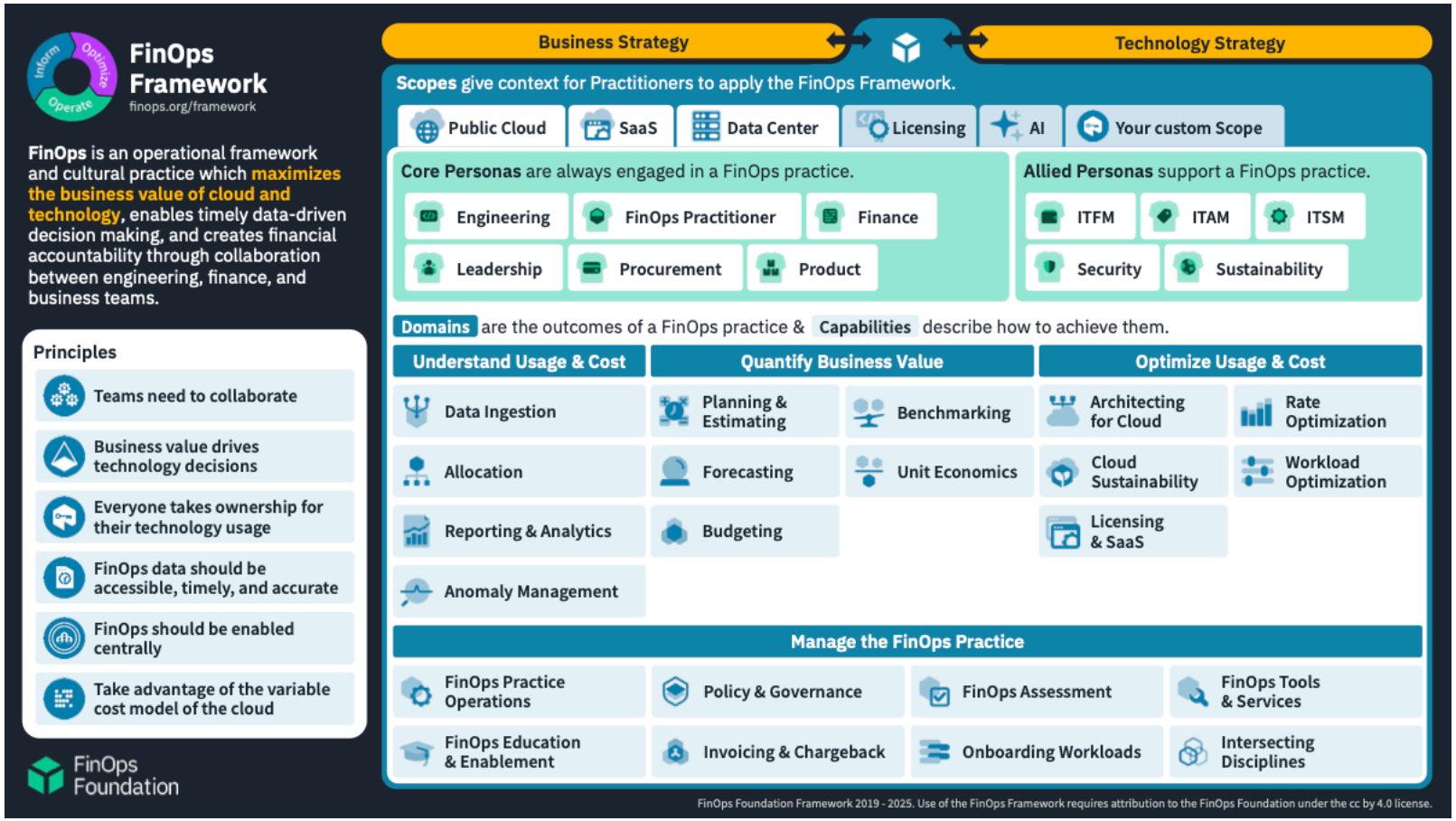
1. Maksymalizacja wartości biznesowej z wydatków na chmurę
2. Współpraca między finansami, inżynierią i biznesem
3. Narzędzia i praktyki do zarządzania kosztami chmury
4. Świadome kompromisy: szybkość vs koszt vs jakość

FinOps

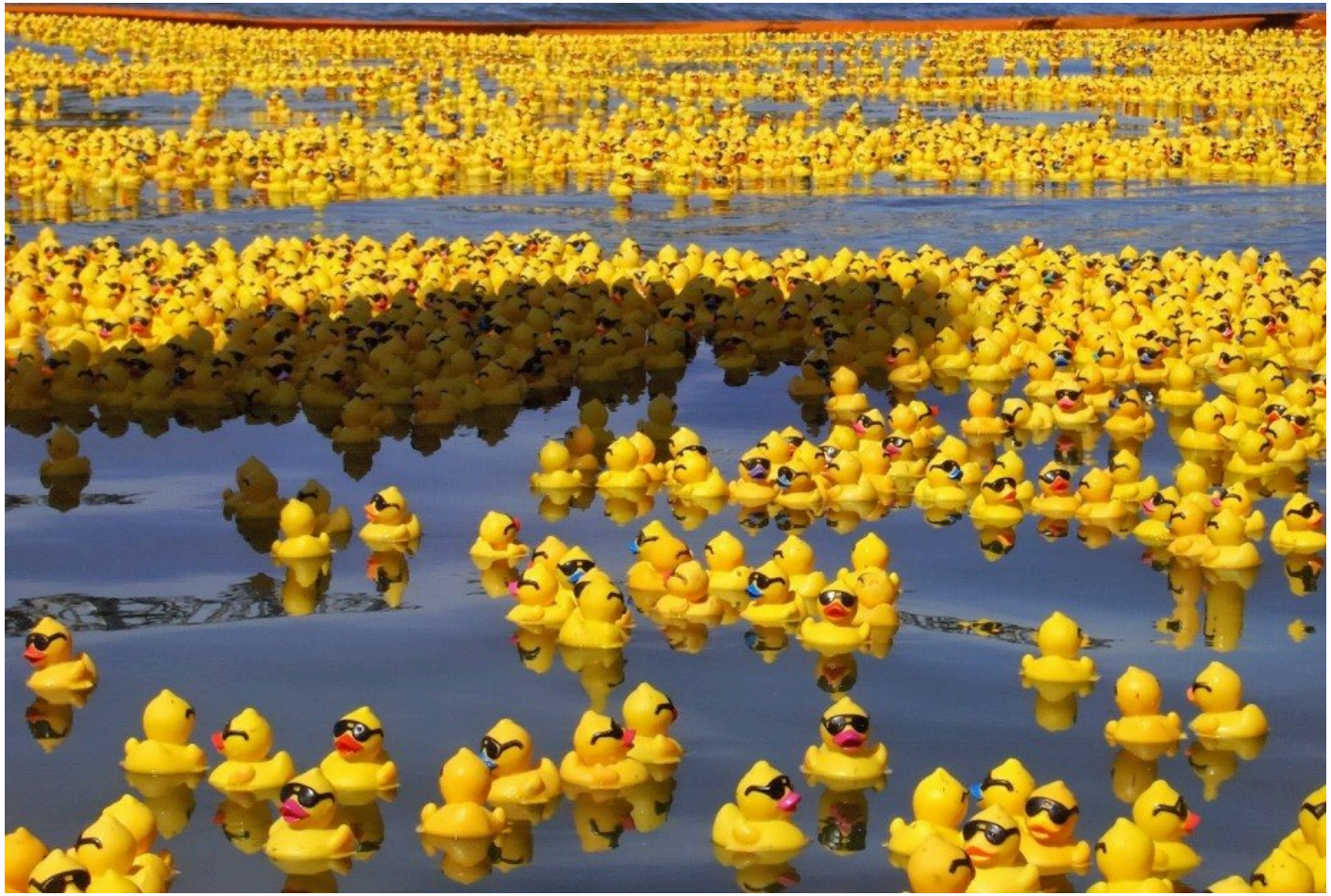
Podobnie jak Lean, czy DevOps, jest to kulturą, pracą w iteracjach, niekoniecznie restrykcyjny proces.

- FinOps Foundation - projekt Linux Foundation (od 2020), 60k+ członków z 15k+ firm, certyfikacje i standardy (FOCUS)
- FinOps Framework - zestaw zaobserwowanych praktyk + mental model dla FinOps oraz rekomendacje

FinOps Framework (à la carte)



Cloud Computing

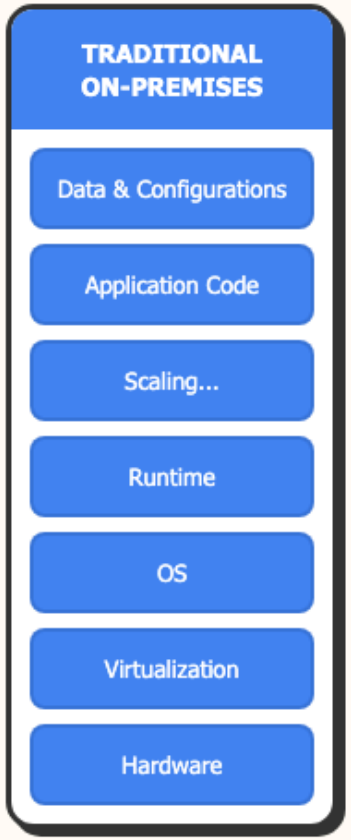


Cloud Computing

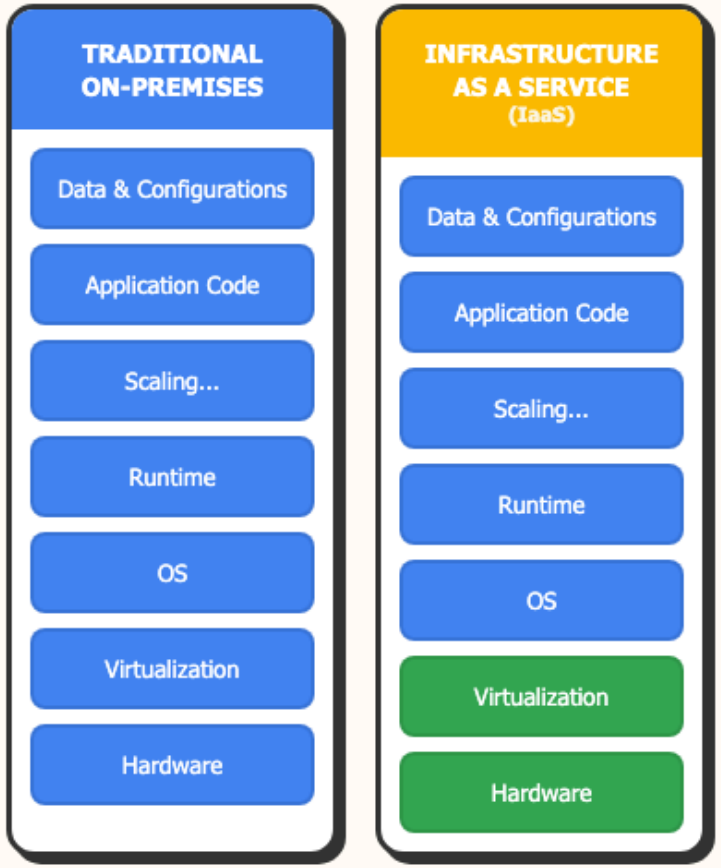
Korzyści:

- Elastyczność i skala
- Przyśpieszenie innowacyjności
- Pay-as-you-go

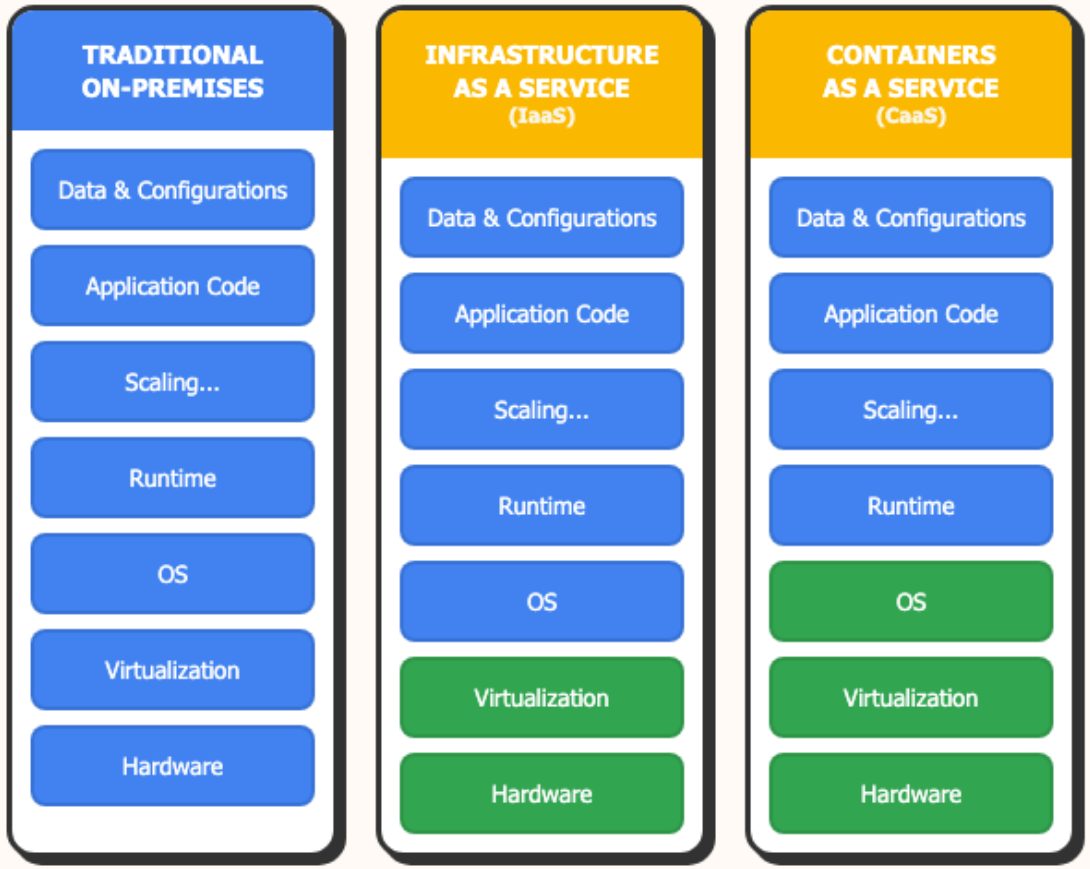
Cloud Computing



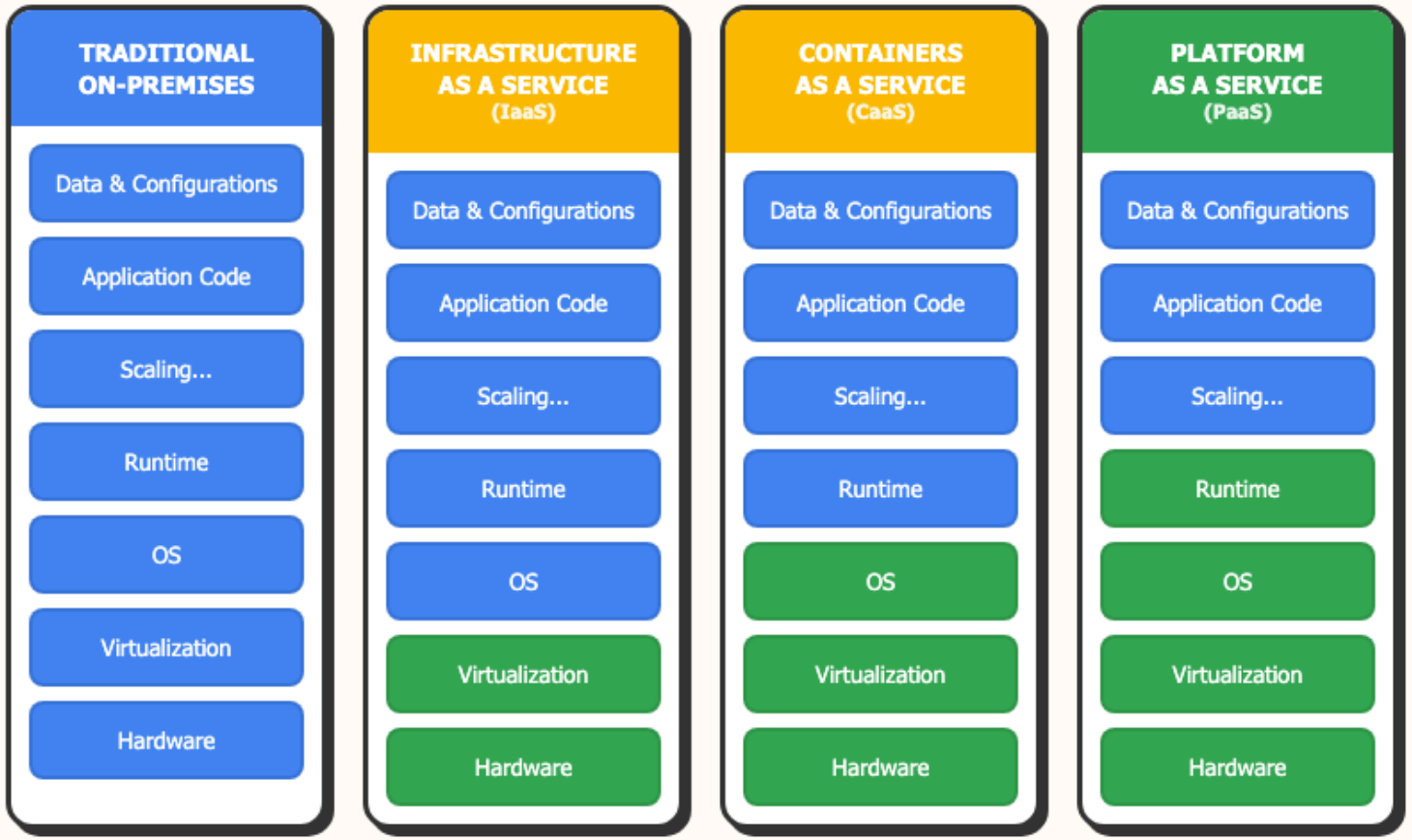
Cloud Computing



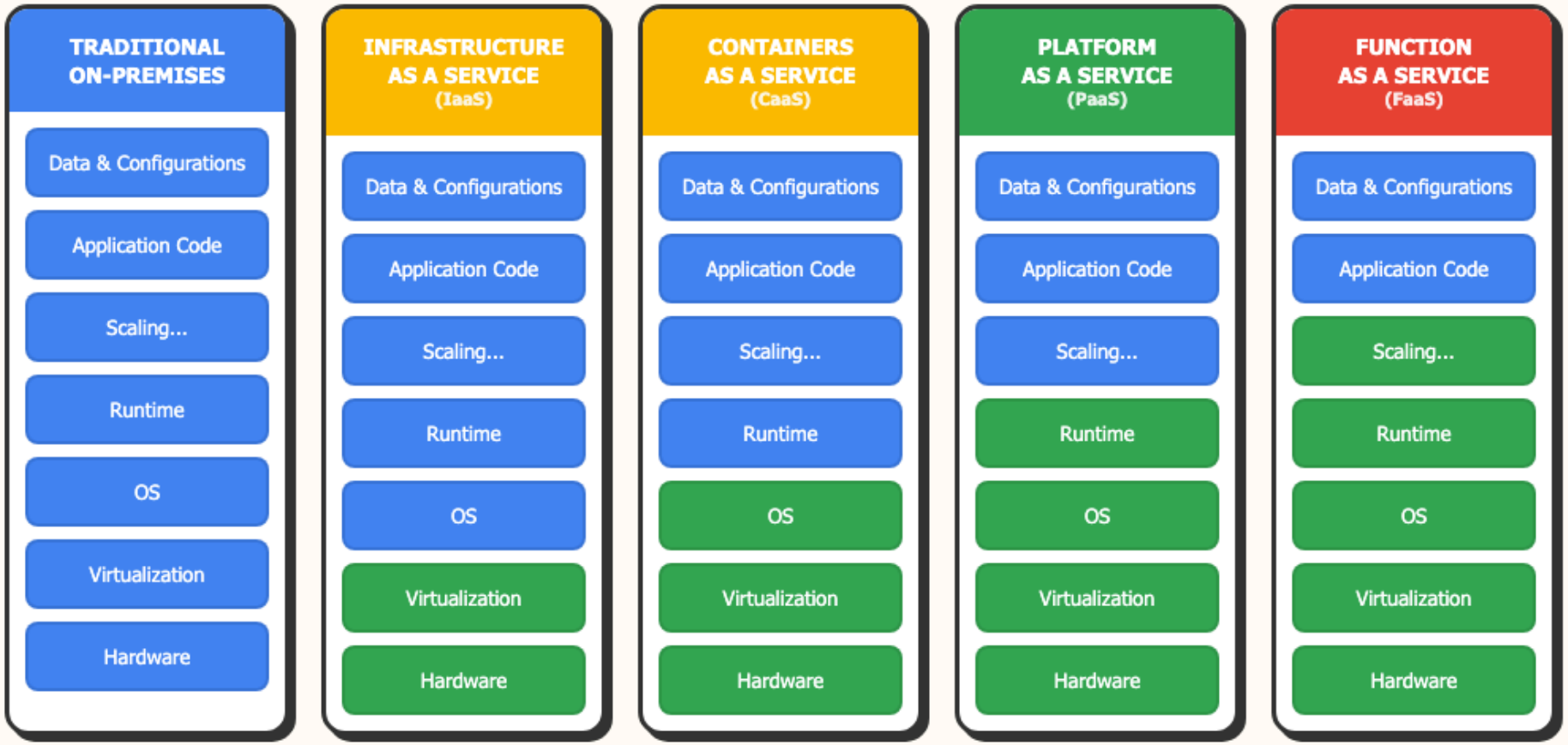
Cloud Computing



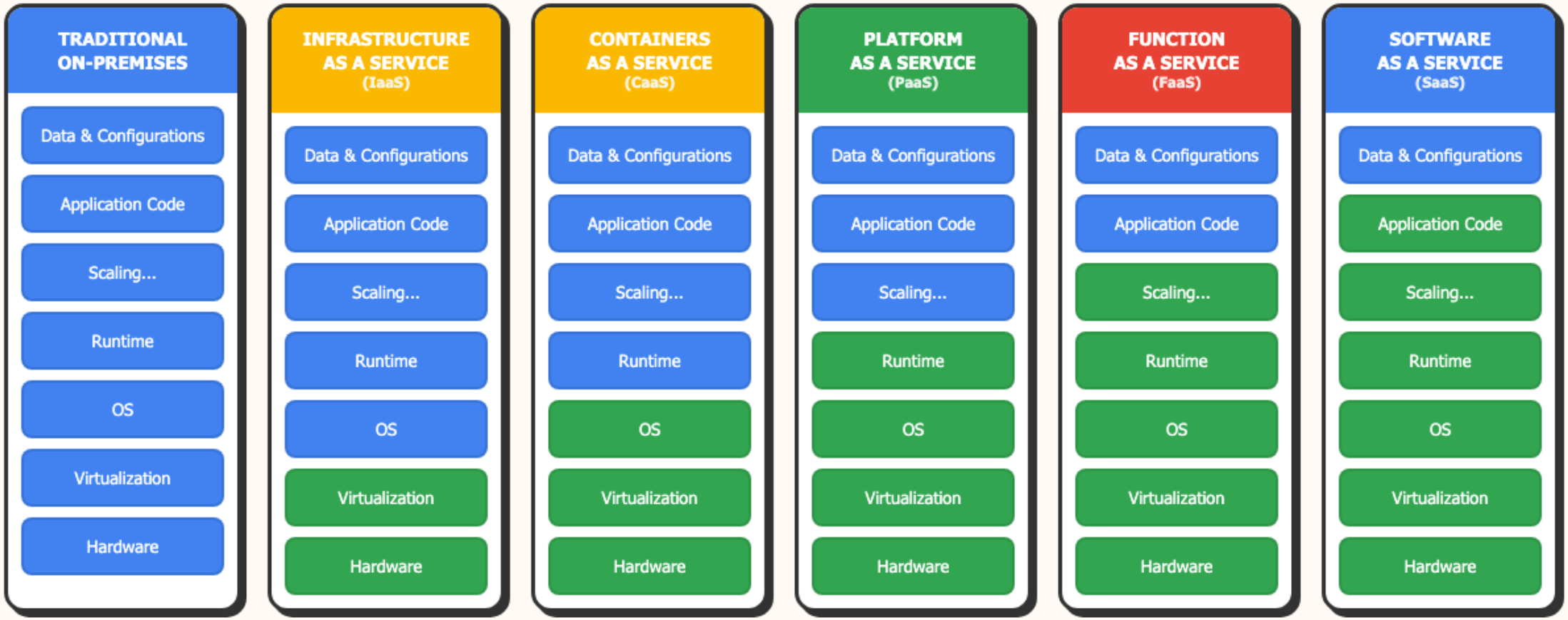
Cloud Computing



Cloud Computing



Cloud Computing



On-Premise

To nie komputery w piwnicy:

- On-Premise → Private Cloud
- Wirtualne Maszyny → Kontenery (CaaS)
- Proprietary → Open Source (KubeVirt, Qemu)

OPEX vs CAPEX

OPEX (Operating Expenses)	CAPEX (Capital Expenses)
Bieżące koszty operacyjne	Inwestycje kapitałowe
Płatność za użycie	Zakup z góry
Elastyczność	Długoterminowe zobowiązanie
Brak amortyzacji	Amortyzacja w czasie
Łatwiejsze skalowanie	Trudniejsze skalowanie
Łatwiej powiązać z biznesem	

Cloud = OPEX model, CAPEX = on-Premise

Pricing - Cloud

- Pay-as-you-go (*h*, ..., *zarządzane zasoby*)
- Reserved Resources
- Spot Resources

Pricing - Cloud

Model	Oszczędności	Charakterystyka
Pay-as-you-go	0%	Elastyczność, brak zobowiązań
Reserved	do 72%	Commitment 1-3 lata
Spot	do 90%	Może zostać przerwany

$$Savings = \sum (C_{on\_demand} - C_{reserved}) - C_{waste}$$

Pricing - On-Premises

W dużych firmach, departamenty rozliczają się w podobny sposób jak to się dzieje w chmurze.

Total Cost of Ownership

Koszty

$$TCO = C_{Infrastruktura} + C_{Ludzie} + C_{Ryzyko}$$

TCO - orientacyjnie

	mała skala	średnia	duża	bardzo duża
OnPrem	🔴	🔴	🟡	🟢*
IaaS	🟡	🟡	✅	✅
CaaS	🟡	✅	✅	✅
FaaS	🟢	✅	🟡	🔴
PaaS	🟢	🟡	🔴	🔴

🟢 niskie · 🟡 średnie · 🔴 wysokie · ✅ optymalne

tablica

TCO - orientacyjnie

1. PaaS dość szybko robi się drogi, vendor-locking!
2. PaaS migruje się na IaaS albo CaaS
3. Możemy przepisać aplikacje na FaaS
4. Nowa aplikacja? Gdzie możemy, używajmy FaaS!

TCO - orientacyjnie

- CaaS i IaaS są obecnie najpopularniejsze.
- Rynek idzie w CaaS i FaaS.

TCO - Grzechy PL/Europa

- buy, build, or wait
- tendencja na *build* lub *wait*
- dotyczy to narzędzi i szkoleń

TCO - Grzechy PL/Europa

1. Czas inżynierów kosztuje, zbudowanie rozwiązania czy konfiguracja Open Source nie jest za darmo
2. ... firmy zaskakuje TCO takiego rozwiązania, np., aktualizacje czy backup.

TCO - Grzechy PL/Europa

- Mniej czasu na innowacje.

TCO - buy, build, or wait

Rekomendacja:

1. Czy to jest core competency? Nie, → delegacja.
2. Proces (dokument) wokół decyzji *buy, build, or wait*
3. Śledzenie TCO rozwiązań in-house i reagowanie na problemy
4. Open Source/Open Core/unikać vendor lockingu.

Koszt vs X

Relacja między kosztami (TCO i budowy/instalacji), a kluczowymi aspektami aplikacji/systemu.

Koszt vs dostępność/SLA

- Wyższa dostępność → większy koszt

tablica

Dostępność

Dostępność:

- **SLA**/SLO/SLI
- **Service Level Agreement**

SLA

SLA	Downtime/rok	Downtime/miesiąc
99.5%	~43h 48min	~3h 39min
99.9%	~8h 46min	~43min 50s
99.95%	~4h 23min	~21min 55s
99.99% (four nines)	~52min 36s	~4min 23s
99.999% (five nines)	~5min 15s	~26s

SLA

Pragmatycznie i systemowo:

- Ile nas kosztuje bycie offline?
- Zarządzanie oczekiwaniami
  (SLA w kontrakcie, komunikacja z klientem)
- Disaster recovery.

SLA

Apetyt na ryzyko (przeczytaj):

- per scenariusz
- zyski vs straty

Koszt vs wydajność

- Większa wydajność → większy koszt
- Pragmatycznie i systemowo
- Testy wydajnościowe

Koszt vs Certyfikaty/Audyty

(Zbyt dosłowne) wymagania certyfikatów → większy koszt

Nie utonąć z Certyfikatami/Audytami

Rozróżnić:

1. Przejście audytu
2. Faktyczne poprawienie bezpieczeństwa i procesów (tutaj inwestujemy!)

AI

Kosztowana technologia

AI - Why i What

The Drivetrain Approach

źródło

The Drivetrain Approach

Krok	Opis	Przykład (wyszukiwarka)
Objective	Zdefiniuj cel biznesowy	Pokazać najbardziej trafne wyniki
Levers	Określ dostępne działania	Ranking wyników wyszukiwania
Data	Zbierz potrzebne dane	Eksperymenty A/B, logi użytkowników
Models	Zbuduj model predykcyjny	Model rankingowy optymalizujący CTR

AI

- Wyścig zbrojeń trwa - Gemini, Anthropic, OpenAI i Mistral
- Testowanie pozwala wybrać najefektwyniejsze ML API
- Monitoring wyników oraz **kosztów**

AI Open Source

- Modele są coraz lepsze – Deepseek czy Kim
- Dla wielu potrzeb, własny model działa (fine-tuned, a nawet bez) i jest o wiele rzędów wielkości tańszy

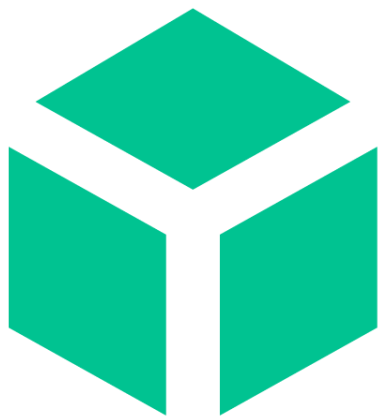
AI

Strategia "Build vs. Buy vs. Fine-tune"

Model wdrożenia	Koszt początkowy	Koszt bieżący	Elastyczność
SaaS / API	Bardzo niski	Wysoki (za token)	Niska
Fine-tuning	Średni	Średni	Wysoka
Custom Build	Bardzo wysoki	Wysoki (Infrastruktura)	Pełna

FinOps

1. Maksymalizacja wartości biznesowej z wydatków na chmurę
2. Współpraca między finansami, inżynierią i biznesem
3. Narzędzia i praktyki do zarządzania kosztami chmury
4. Świadome kompromisy: szybkość vs koszt vs jakość



FinOps
Foundation

FinOps is an operational framework and cultural practice which ***maximizes*** *the* ***business value*** *of cloud and technology, enables timely data-driven decision making, and creates* ***financial accountability*** *through collaboration between engineering, finance, and business teams.*

FinOps Framework

finops.org/framework

FinOps is an operational framework and cultural practice which **maximizes the business value of cloud and technology**, enables timely data-driven decision making, and creates financial accountability through collaboration between engineering, finance, and business teams.

Principles

- Teams need to collaborate
- Business value drives technology decisions
- Everyone takes ownership for their technology usage
- FinOps data should be accessible, timely, and accurate
- FinOps should be enabled centrally
- Take advantage of the variable cost model of the cloud

FinOps Foundation

Business Strategy ↔ **Technology Strategy**

Scopes give context for Practitioners to apply the FinOps Framework.

Public Cloud | SaaS | Data Center | Licensing | AI | Your custom Scope

Core Personas are always engaged in a FinOps practice.

- Engineering
- FinOps Practitioner
- Finance
- Leadership
- Procurement
- Product

Allied Personas support a FinOps practice.

- ITFM
- ITAM
- ITSM
- Security
- Sustainability

Domains are the outcomes of a FinOps practice & **Capabilities** describe how to achieve them.

Understand Usage & Cost

- Data Ingestion
- Allocation
- Reporting & Analytics
- Anomaly Management

Quantify Business Value

- Planning & Estimating
- Benchmarking
- Forecasting
- Unit Economics
- Budgeting

Optimize Usage & Cost

- Architecting for Cloud
- Rate Optimization
- Cloud Sustainability
- Workload Optimization
- Licensing & SaaS

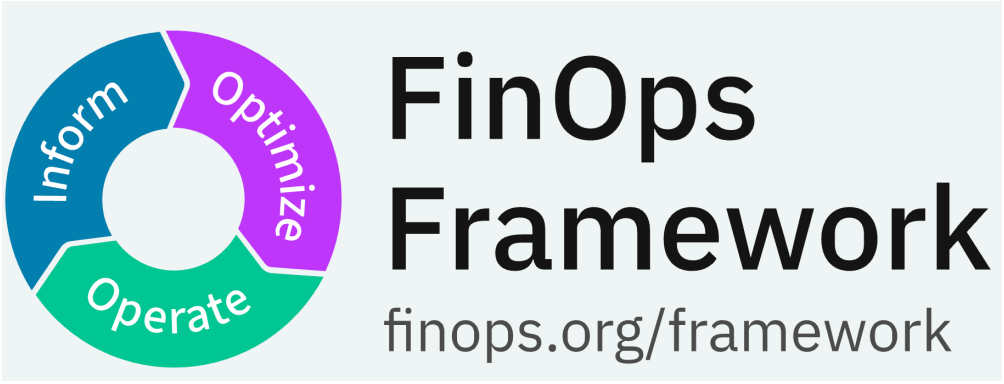
Manage the FinOps Practice

- FinOps Practice Operations
- Policy & Governance
- FinOps Assessment
- FinOps Tools & Services
- FinOps Education & Enablement
- Invoicing & Chargeback
- Onboarding Workloads
- Intersecting Disciplines

FinOps Process

Fazy:

1. Inform - Widoczność i alokacja
2. Optimize - Optymalizacja stawek i zużycia
3. Operate - Operacjonalizacja i kultura



Najważniejszy element

Jasny i regularny przekaz od liderów – leadership team, all-hands, liderów tech dla zespołów:

- gdzie jesteśmy z biznesem (*why*)
- koszta vs inwestycje vs security vs ryzyko per product/unit (*what*)
- Opcjonalnie: constains (X is our partner)

Najważniejszy element

Mało efektywne:

1. CEO: Chcemy obniżyć koszty o 25%
2. CTO: Ok, czyli mamy zwolnić tempo na projekcie A i B
3. CEO: Nie, mamy dowieźć A i B i znaleźć oszczędności
4. CTO: To może przesunę ludzi z B
5. CEO: A i B są kluczowymi projektami

Sposób budowy aplikacji / Product Mgmt

Tracer-bullet development:

- Szybkie zbudowanie działającej funkcjonalności do pokazania klientowi
- Iteracyjne dodawanie funkcjonalności

Proces decyzyjny

- Dokument opisujący problem i kontekst
- Kilka opcji z analizą trade-off
- Jasne success criteria dla rozważanych rozwiązań

Inform - Widoczność i alokacja

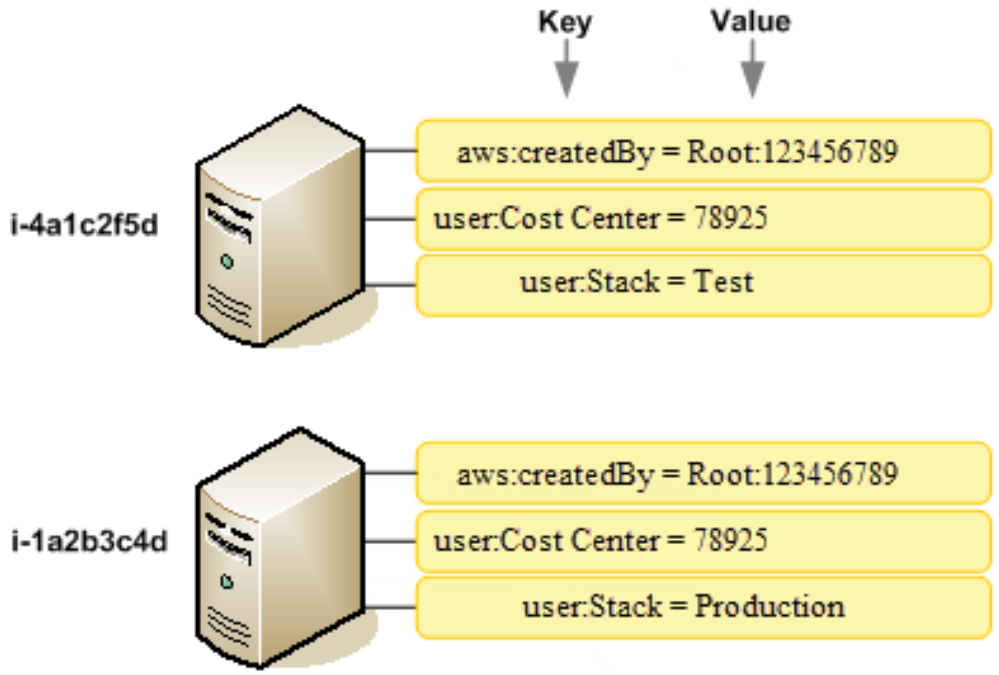
Większość organizacji boryka się z brakiem przejrzystości wydatków. Faza "Inform" polega na stworzeniu precyzyjnej mapy konsumpcji zasobów.

Inform - Widoczność i alokacja

1. Tagowanie i alokacja:
   Przypisanie zasobow do centrum kosztowego, projektu lub produktu
2. Jednostkowa ekonomika (Unit Economics):
   koszt IT → kosztu na jednostkę biznesową,
3. Prognozowanie:
   Wykorzystanie danych historycznych i algorytmów ML

Inform - Widoczność i alokacja

Przykład dla Amazona (dokumentacja AWS):



Inform - Widoczność i alokacja

Na początku, moze to być arkusz, ktory uzupełniamy co miesiąc:

1. Nazwa usługi, np. Datadog
2. What i Why?
3. Koszt na miesiąc

wraz z pierwszym wyliczeniem *Unit Economics*.

Inform - Unit Economics

- Metryka łącząca biznes z kosztem IT
- Wspólny język łączący Inżynierię, Finanse oraz Biznes
- Pozwala wcześnie zauważyć trendy (zagrożenia i pozytywne)
- Ułatwia dyskusję o kosztach

Inform - Unit Ecomomics

- Idealny świat: łatwo powiązać koszt ze sprzedażą
  Zazwyczaj: aproksymacja
- Koszt za transakcję, na klienta, kampanię marketingową, itp.

Inform - Unit Ecomomics

	Luty	Marzec	Kwiecień
Datadog	1000	1500	2000
AWS	4000	5000	5500
Anthropic	3000	4000	6500
Total	8000	10500	14000
Transakcji	100	120	130
Unit Economics	80	87.5	107.7

Inform - Unit Economics

Pre-Crawl

No Cost Models

- Not started any cost models

Crawl

Unit Cost

- Blended unit cost based on the invoice.
- Unblended Unit Cost based on contract pricing.

Walk

Unit Cost Integration

- Unit cost used to convert unit forecast to budget.
- Usage and cost conversations

Run

Unit Economics

- Cost to produce
- Cost to serve a customer
- Cost to serve an application
- Cost to serve a file

Inform - Prognozowanie

Podejścia:

1. **Historyczne** - ekstrapolacja z Cost & Usage data
2. **Trend-based** - analiza wzorców wzrostu
3. **Driver-based** - powiązanie z metrykami biznesowymi (np. liczba użytkowników)

Inform - Prognozowanie

Best practices:

- Regularne aktualizacje (co miesiąc/kwartał)
- Współpraca: Finance + Engineering + Product
- Alerty przy przekroczeniu progów
- Uwzględnienie rabatów (RI, Savings Plans)

KPI: Forecast Variance (cel: Crawl ±20%, Walk ±15%, Run ±12%)

Inform - FOCUS

FinOps Open Cost & Usage Specification, uniwersalny format:

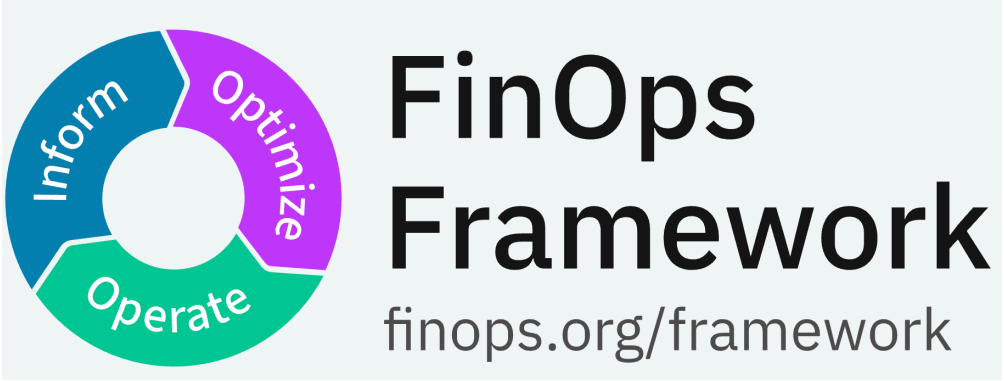
- Format
- Przyklad

więcej na focus.finops.org

Inform - SaaS

- Koszty SaaS rosną szybciej niż infrastruktura
- Shadow IT: niekontrolowane subskrypcje na kartach kredytowych
- Rozwiązanie: współpraca z Finance + narzędzia typu Brex/Ramp

Optimize - Optymalizacja stawek i zużycia



Optimize - Optymalizacja stawek i zużycia

- Zarządzanie niepotrzebnymi zasobami - automatyzacja srodowisk testowych oraz skalowania
- Rightsizing & consolidation - dopasowanie wielkości maszyn do ich faktycznego obciążenia
- Optymalizacja stawek (Rate Optimization) - wykorzystanie saving plans, license management, itp.
- Migracje i transformacje

Optimize - Nieuzywane zasoby

- Praktycy: zacznij od zapomnianych zasobów, early (big) wins w każdym projekcie FinOps
- Automatyczne usuwanie nieużywanych zasobów:
  - środowiska deweloperskie
  - produkcja w weekend, itp. itd.

Optimize - Rightsizing i Konsolidacja

- Czy naprawde jest to nam potrzebne? Dane przed 2 lat w bazie danych.
- Czy koszt jest warty danej funkcjonalności?

Optimize - Rightsizing i Konsolidacja

- Pragmatycznie i systemowo (ryzyko, nagły wzrost ruchu)
- Wymaga danych z procentowego obciążenia
- Konsolidacja - więcej niż jedna aplikacja na tej samej maszynie

Niektóre technologie łatwo zoptymalizować, inne są trudniejsze.

Optimize - Optymalizacja stawek

Chmura:

1. Reserved instances - 30%+ taniej
2. Spot instances - do rzędu wielkości mniejsze koszta
3. Frame-contracts

Optimize - Optymalizacja stawek

SaaS/PaaS:

- Najlepsze wyniki (20% - 50% oszczędności), widziałem, zatrudniając zewnętrzną firmę do zarządzania licencjami i negocjowania kontraktów.

Optimize - Migracje i Transformacja

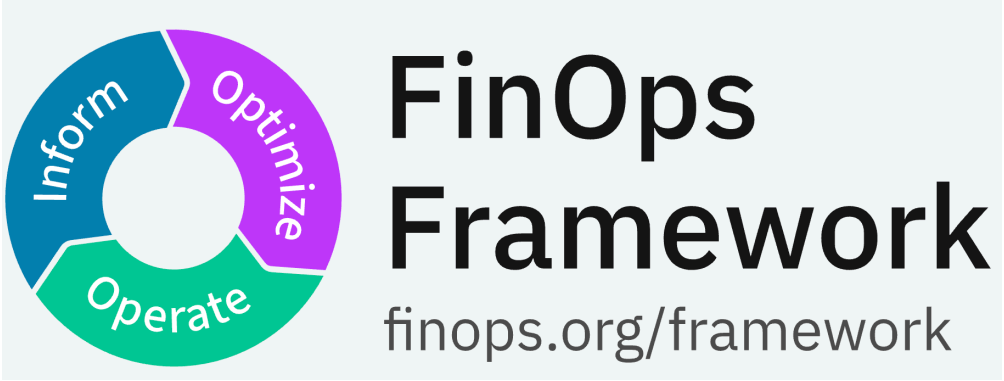
- Przeniesienie mniej krytycznych elementów systemu na tańszego dostawcę
- Przejście na technologie oparte o Open Source (Open/Close Core) z mniejszym ryzykiem vendor-lockin
- Uwage na Value Leakage

Optimize - Migracje - Case Study

Swisscom (artykuł)
migracja z VMware do Kubermatic. Korzyści:

- niższa licencja
- future proof: rozwiązanie oparte o Open Source/Cloud Native
- rozwiązanie z API, zamiast ClickOps first

Operate - Operacjonalizacja i kultura



Operate - Operacjonalizacja i kultura

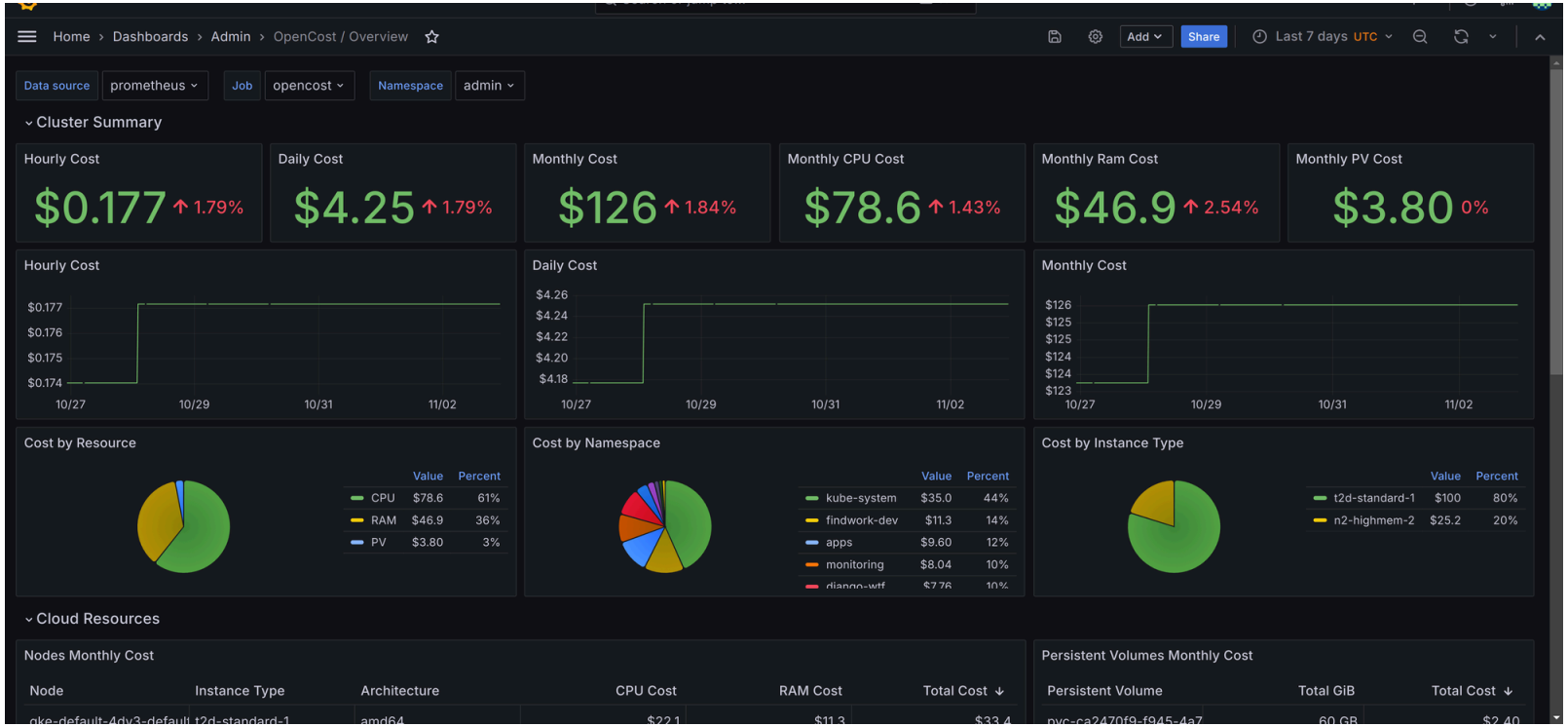
- Demokratyzacja wiedzy o kosztach, łatwy dostęp do metryk FinOps, monitoring kosztów
- Wprowadzenie polityk i reguł (governance)
- Shift-left, automatyzacja i guardrails
- Regularny przegląd kosztów zarówno na poziomie firmy, jak i zespołów technicznych

Operate - demokratyzacja

- Trudno wymagać, aby organizacja zarządzała czym, czego nie widzą
- Dlatego dashboardy dla inżynieringu powinny mieć metryki:
  - dostarczaniem wartości klientowi
  - zarobionymi środkami
  - koszta / unit economics

Operate - demokratyzacja

Dostęp do metryk i alerty:

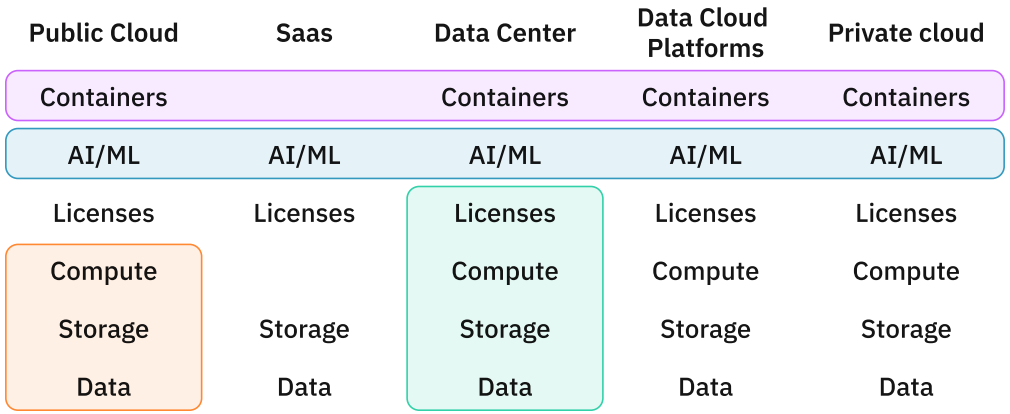


Operate - polityki

1. Utworzenie working group dla FinOps na poziomie org technicznej i firmy
2. Tworzenie polityk i propozycji dla ich automatyzacji

FinOps Scopes

Odpowiedzialność zespołu oraz poszczególnych osób (**Personas**) współpracujących w kontekście FinOps



Operate - Shift-left

Shift-left w FinOps to wcześniejsze uwzględnianie kosztów w cyklu życia oprogramowania.

Operate - Shift-left

1. Koszty w decyzjach architektonicznych i projektowych już na starcie.
2. Automatyczne alerty i limity kosztów podczas developmentu i testów.
3. Szkolenie zespołów technicznych z optymalizacji kosztów chmury.
4. Narzędzia i polityki FinOps w CI/CD i procesach developerskich.

Operate - Shift-left i automatyzacja

- Konfiguracja polityki dostępu per aplikacja i per team
- Policy-as-Code (OPA):
  - wymaganie tagowania: przykład
  - niewdozwolone zasoby: przykład
- Automatyczne usuwanie zasobów
- Alerty i automatyczne raporty

Operate - Regularny przegląd kosztów

1. Zespół techniczny (nie koniecznie cały, ale przynajmniej 2 osoby)
2. Unit cost, zmiany w ruchu, częścią prezentacji, np., na all-hands
3. Spotkania 1 na jeden lub dwa miesiące z działem finansowym

Operate - Shift Left

- Hackathony / challenges - Nagroda za znalezienie oszczędności
- Regularne sprinty (1 na miesiąc lub 2), gdzie spłacamy dług niepotrzebnych kosztów

Operate - Shift Left

Koszty chmury powinny być analizowane tak jak inne aspekty jakości oprogramowania:

- **Pre-mortem** — przewidywanie kosztów przed wdrożeniem
- **Post-mortem** — analiza anomalii kosztowych po incydencie

Operate - Broken Ownership

- Częsty problem po wprowadzeniu DevOps
- DevOps/Platform mimo przekazania zasobu, jest ciągle za niego odpowiedzialny

Operate - Broken Ownership

Zespoły konsumujące zasoby powinny:

- mieć dostęp do danych operacyjnych
- zasoby powinny aktywnie brać udział ich optymalizacji

FinOps Framework

Faza FinOps	Kluczowe działanie	Metryka sukcesu	Narzędzia
Inform	Widoczność 100% wydatków	% alokowanych kosztów	Dashboards, Tagi
Optimize	Redukcja marnotrawstwa	% oszczędności (Savings %)	Rightsizing, Spot
Operate	Zmiana kultury pracy	Dokładność prognoz (+/- 5%)	Polityki, Automatyzacja

FinOps Framework

finops.org/framework

FinOps is an operational framework and cultural practice which **maximizes the business value of cloud and technology**, enables timely data-driven decision making, and creates financial accountability through collaboration between engineering, finance, and business teams.

Principles

- Teams need to collaborate
- Business value drives technology decisions
- Everyone takes ownership for their technology usage
- FinOps data should be accessible, timely, and accurate
- FinOps should be enabled centrally
- Take advantage of the variable cost model of the cloud

FinOps Foundation

Business Strategy ↔ **Technology Strategy**

Scopes give context for Practitioners to apply the FinOps Framework.

Public Cloud | SaaS | Data Center | Licensing | AI | Your custom Scope

Core Personas are always engaged in a FinOps practice.

- Engineering
- FinOps Practitioner
- Finance
- Leadership
- Procurement
- Product

Allied Personas support a FinOps practice.

- ITFM
- ITAM
- ITSM
- Security
- Sustainability

Domains are the outcomes of a FinOps practice & **Capabilities** describe how to achieve them.

Understand Usage & Cost

- Data Ingestion
- Allocation
- Reporting & Analytics
- Anomaly Management

Quantify Business Value

- Planning & Estimating
- Benchmarking
- Forecasting
- Unit Economics
- Budgeting

Optimize Usage & Cost

- Architecting for Cloud
- Rate Optimization
- Cloud Sustainability
- Workload Optimization
- Licensing & SaaS

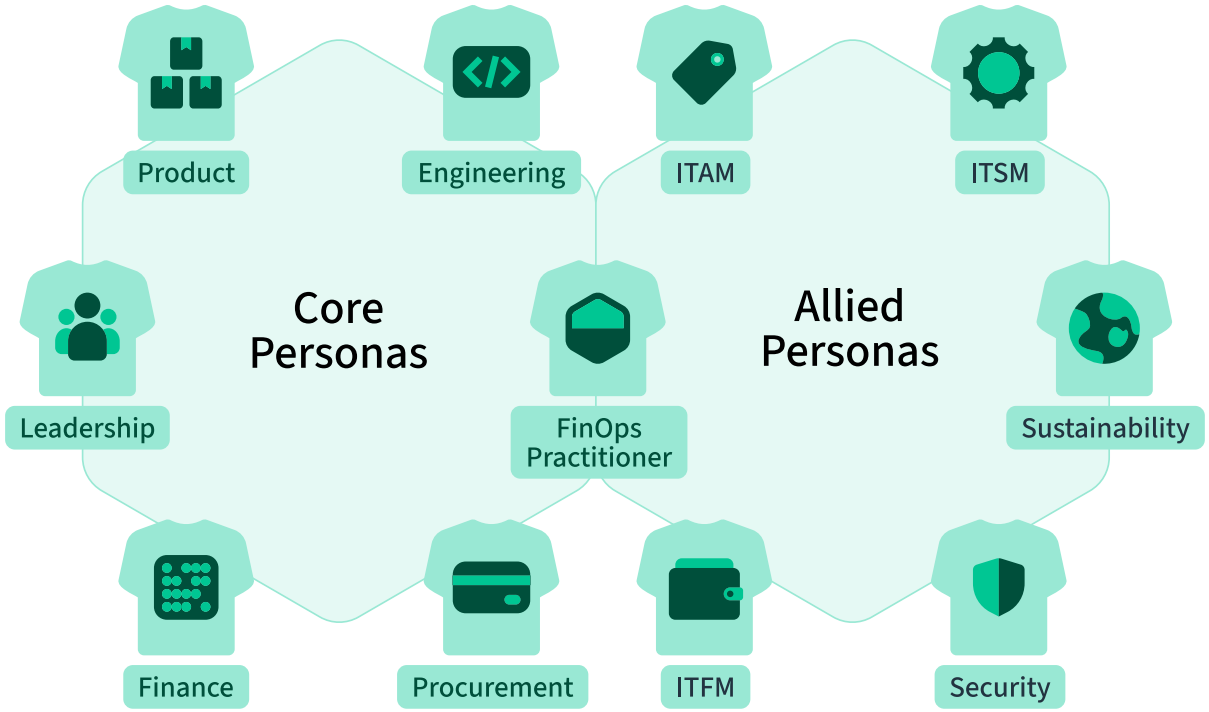
Manage the FinOps Practice

- FinOps Practice Operations
- Policy & Governance
- FinOps Assessment
- FinOps Tools & Services
- FinOps Education & Enablement
- Invoicing & Chargeback
- Onboarding Workloads
- Intersecting Disciplines

FinOps Principles

1. Teams need to collaborate
2. Business value (not units) drives technology
3. Everyone takes ownership of their cloud usage
4. Timely FinOps reports
5. Centralized FinOps team drives best practices
6. Leverage the variable cost model of the cloud

FinOps Personas



finops.org

FinOps Domains

Domeny grupują powiązane capabilities (zdolności) w 4 obszary:

1. **Understand** - zbieranie danych o kosztach i użyciu
2. **Quantify** - łączenie kosztów z wartością biznesową
3. **Optimize** - redukcja kosztów i marnotrawstwa
4. **Manage** - governance, edukacja, procesy

FinOps Domain: Understand Usage & Cost

Zbieranie informacji o kosztach, użyciu i metrykach wydajności chmury.

- Data Ingestion
- Allocation (tagowanie, hierarchia)
- Reporting & Analytics
- Anomaly Management

FinOps Domain: Quantify Business Value

Łączenie danych o kosztach z wartością biznesową.

- Planning & Estimating
- Forecasting
- Budgeting
- Benchmarking
- Unit Economics

FinOps Domain: Optimize Usage & Cost

Zapewnienie, że zasoby są używane tylko gdy przynoszą wartość i kupowane po najniższej akceptowalnej cenie.

- Architecting for Cloud
- Rate Optimization
- Workload Optimization
- Cloud Sustainability
- Licensing & SaaS

FinOps Domain: Manage the FinOps Practice

Ciągłe doskonalenie organizacji poprzez ludzi, procesy i technologię.

- FinOps Practice Operations
- Policy & Governance
- FinOps Education & Enablement
- Invoicing & Chargeback
- Intersecting Disciplines

FinOps Maturity Model

Model **Crawl → Walk → Run** pozwala organizacjom stopniowo rozwijać dojrzałość FinOps.

Nie wszystkie capability muszą być na poziomie "Run" — **wartość biznesowa** powinna decydować o priorytetach.

Maturity: Crawl

Aspekt	Opis
Raportowanie	Minimalna infrastruktura narzędziowa
Procesy	Podstawowe polityki ustalone
Adopcja	Zrozumienie istnieje, brak szerokiej adopcji
Cel	Identyfikacja "low hanging fruit"

Metryki: ≥50% alokacji kosztów, ~60% commitment coverage, ±20% forecast variance

Maturity: Walk

Aspekt	Opis
Raportowanie	Automatyzacja pokrywa większość wymagań
Procesy	Konsekwentnie stosowane w organizacji
Adopcja	Edge cases zidentyfikowane i analizowane
Cel	Średnie do wysokich targety wydajności

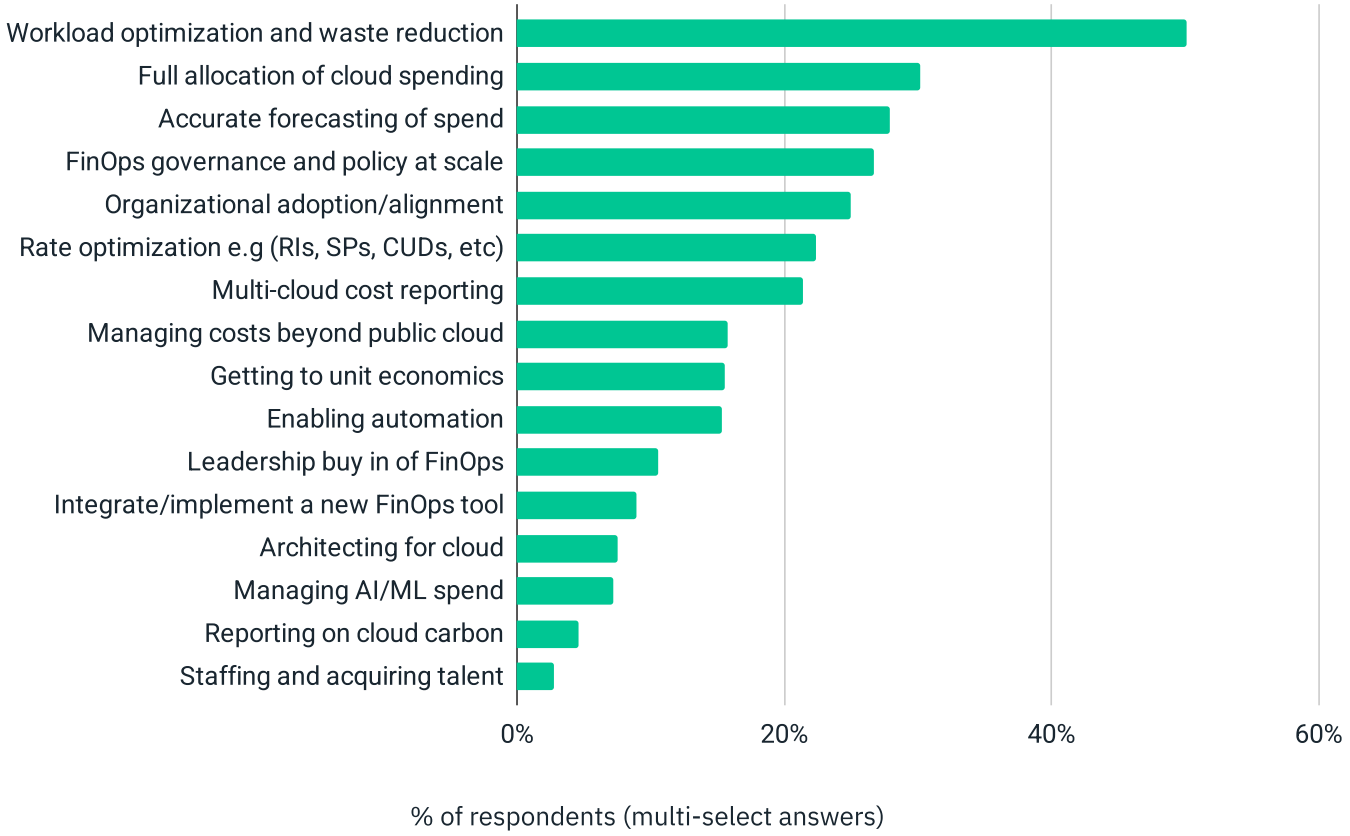
Metryki: ≥80% alokacji kosztów, ~70% commitment coverage, ±15% forecast variance

Maturity: Run

Aspekt	Opis
Raportowanie	Automatyzacja jako standard
Procesy	Trudne edge cases aktywnie rozwiązywane
Adopcja	Pełne zrozumienie we wszystkich zespołach
Cel	Bardzo wysokie targety wydajności

Metryki: >90% alokacji kosztów, ~80% commitment coverage, ±12% forecast variance

State Of FinOps



data.finops.org

Kluczowe zalecenia

- **Integracja** - koszt = first-class citizen w cyklu rozwoju produktu
- **Widoczność** - bez tagowania i alokacji nie ma skutecznego FinOps
- **Elastyczność** - unikaj długich kontraktów, technologia się zmienia

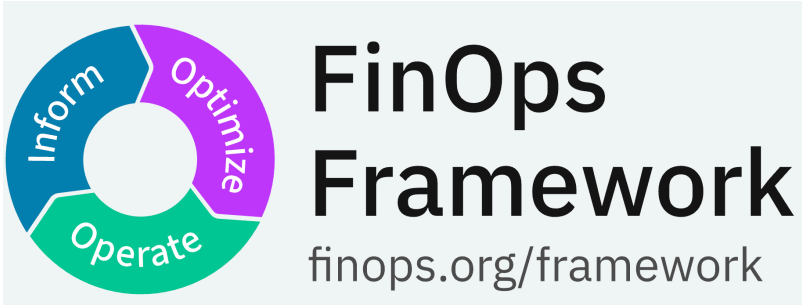
Kluczowe zalecenia

- **Kultura** - FinOps = 20% technologii + 80% ludzi
- **Iteracja** - balans kosztów vs innowacji zmienia się w czasie
- **Dedykacja** - potrzebne są dedykowane osoby lub czas

Kluczowe zalecenia

- FinOps framework
  praktyki, wspólny metryk, oraz model referencyjny

Pytania?



github.com/wojciech11