IN2000
Software Engineering og Prosjektarbeid
Vår 2024

Sikker systemutvikling

Audun Jøsang

Oversikt

- Innebygd informasjonssikkerhet
  - *Security Development Lifecycle*
  - Trusselmodellering og STRIDE
  - Applikasjonssikkerhet og OWASP Top 10
  - Sikkerhet i skyen

Innebygd informasjonssikkerhet og personvern

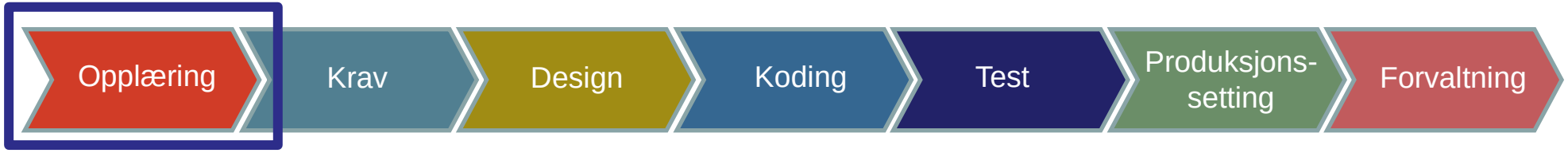
- «Innebygd» informasjonssikkerhet og personvern betyr at det tas eksplisitt hensyn til informasjonssikkerhet og personvern i hele livssyklusen til programvare og applikasjoner.
- Et viktig mål er å finne og redusere sårbarheter tidlig i utviklingsprosessen slik at det blir færre hendelser og sårbarheter å håndtere under drift.
- Microsoft så dette behovet tidlig og har vært ledende her og utviklet *Microsoft Security Development Lifecycle (SDL)*
  - Denne er nå en integrert del av programvareutviklingsprosessen hos Microsoft (og andre)
- Vi kan beskrive livssyklusen som en prosess av 7 faser:

Fase 1: Opplæring

Alle som deltar i utvikling og drift av digitale tjenester og applikasjoner skal ha basiskunnskap og forståelse for

- Informasjonssikkerhet og personvern
- Risikovurderinger
- Trusselmodellering

Dette er egentlig en selvfølge. Det ville være uforsvarlig å utdanne bygningsarkitekter og ingeniører uten å gi dem kunnskap om branntrygghet, fordi arkitekter og ingeniører da ville bygget brannfeller inn i våre bygninger. På samme måte er det uforsvarlig å utdanne informatikere og dataingeniører uten obligatoriske kurs om informasjonssikkerhet, fordi de ferdigutdannede da nødvendigvis ville bygget en sårbar IKT-infrastruktur.



Krystallklare signaler fra myndighetene

- Alle må ha kunnskap om digital sikkerhet.

… men det har ikke alltid vært slik

- Nasjonal strategi fra 2012 nevnte ikke opplæring i informasjonssikkerhet med et eneste ord.
- Stortingsmelding 18-2014 foreslo å spare penger ved at små fagområder kun skulle undervises ved ett lærested.
- Utdanningsdirektoratet forslo i 2014 at kun ett lærested skulle undervise informasjonssikkerhet.
- Fagmiljøene sa tydelig fra at informasjonssikkerhet må undervises overalt.

Endret visjon om IKT-sikkerhet i utdanningen

2014

UDir (Utdanningsdirektoratet) og NOU 2014:5 SAK (Samarbeid, Arbeidsdeling og Konsentrasjon i høyere utdanning):
IT-sikkerhet skal bare undervises ved ett eneste lærested !

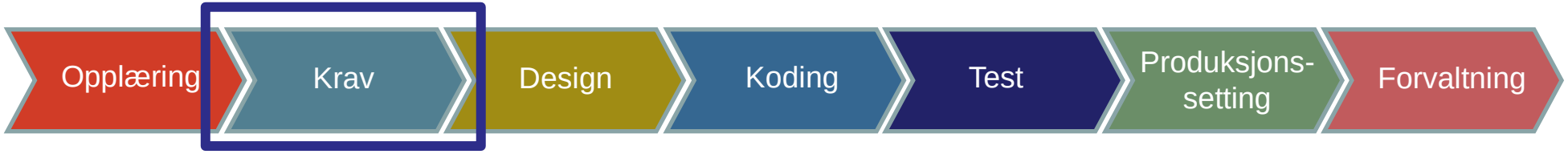
2019

Nasjonal strategi for digital sikkerhetskompetanse 2019:

Digital sikkerhet skal undervises overalt !

Fase 2: Krav om informasjonssikkerhet og personvern

- Kilder til krav om informasjonssikkerhet og personvern:
  1. Krav som følger av god praksis for adekvat sikkerhet i applikasjoner og forretningsprosesser.
  2. Krav om å begrense sikkerhetsrisiko til et akseptabelt forsvarlig nivå.
  3. Juridiske lovbestemte, regulatoriske og kontraktsmessige krav til informasjonssikkerhet og personvern.
- Sikkerhetskrav må kontinuerlig oppdateres for å gjenspeile endringer i nødvendig funksjonalitet, trusselslandskap, lover, forskrifter, reguleringer,…
- Bør gjøres tidlig i utviklingsløpet (innledende design- og planleggingsfasen)
- OWASP Top 10 / ASVS definerer beste praksis for krav om applikasjonssikkerhet



OWASP

- Open Web Application Security Project (OWASP)
  - Ideell organisasjon med mål om å forbedre sikkerheten til applikasjoner og tjenester på internett
  - Gjennom råd, veiledning og verktøy
  - Involverer bedrifter, utdanningsinstitusjoner og enkeltpersoner fra hele verden
  - Flere parallelle prosjekter
- OWASP ASVS (Application Security Verification Standard)
  - Standard med mål om å definere beste praksis for sikker utvikling og testing
- OWASP Top 10
  - Rangerer de 10 mest kritiske sikkerhetsrisikoene for nettapplikasjoner
  - Gir råd om hvordan sårbarhet og risiko kan reduseres
  - Oppdateres med noen års mellomrom
  - Siste revisjon kom ut i 2021, forrige versjon fra 2017
  - Kan være en veldig god kilde for å komme i gang med trusselmodellering

OWASP Top 10

1. **Brudd på tilgangskontroll**
   - Angripere utnytter feil i hvordan tilgangskontroll er håndhevet
   - Kan f.eks. være å lese eller endre andre brukeres data
2. **Kryptografiske feil**
   - Feil relatert til krypto som ofte medfører at sensitiv data eksponeres eller kompromittering av system
3. **Injeksjon**
   - Manipulert input-data sendes til en applikasjon som en del av en forespørsel/kommando som lurer applikasjonen til å utføre utilsiktede eller uautoriserte handlinger
   - Flere varianter, inkludert SQL-injeksjon
   - Cross-site scripting (XSS) er (i 2021 versjonen) en del av denne kategorien
4. **Usikkert design**
   - Risiko relatert til feil i design
5. **Feilkonfigurert sikkerhet**
   - Risiko som skyldes feil i konfigurering
   - F.eks. usikker standardkonfigurering som ikke endres, feilmeldinger som avslører sensitiv informasjon

OWASP Top 10

6. **Sårbare og utdaterte komponenter**
   - Utnyttelse av sårbar/utdatert komponent
   - F.eks. gjennom bruk av (eksternt) bibliotek eller programvaremodul som kjører med samme privileger som applikasjonen hvor de brukes
7. **Feil i identifisering og autentisering**
   - Sjekk av brukers identitet, autentisering eller styring av økt er ofte implementert feil
   - F.eks. dårlige og standard passord, reset av passord, økt-identifikator i URL, …
8. **Feil i (data og programvare) integritet**
   - F.eks. applikasjon bruker modul fra ukjent kilde eller auto-oppdatering uten god nok sjekk av integritet
9. **Utilstrekkelig logging og overvåking**
   - Medfører at man ikke kan detektere og håndtere brudd
10. **Server side request forgery (SSRF)**
   - Angriper får tjener-applikasjon til å gjøre forespørsel til et domene spesifisert av angrep
   - F.eks. kan det medføre at angriper kan lese/oppdatere interne ressurser.

Fase 3: Sikker design, og Fase 4: Sikker koding

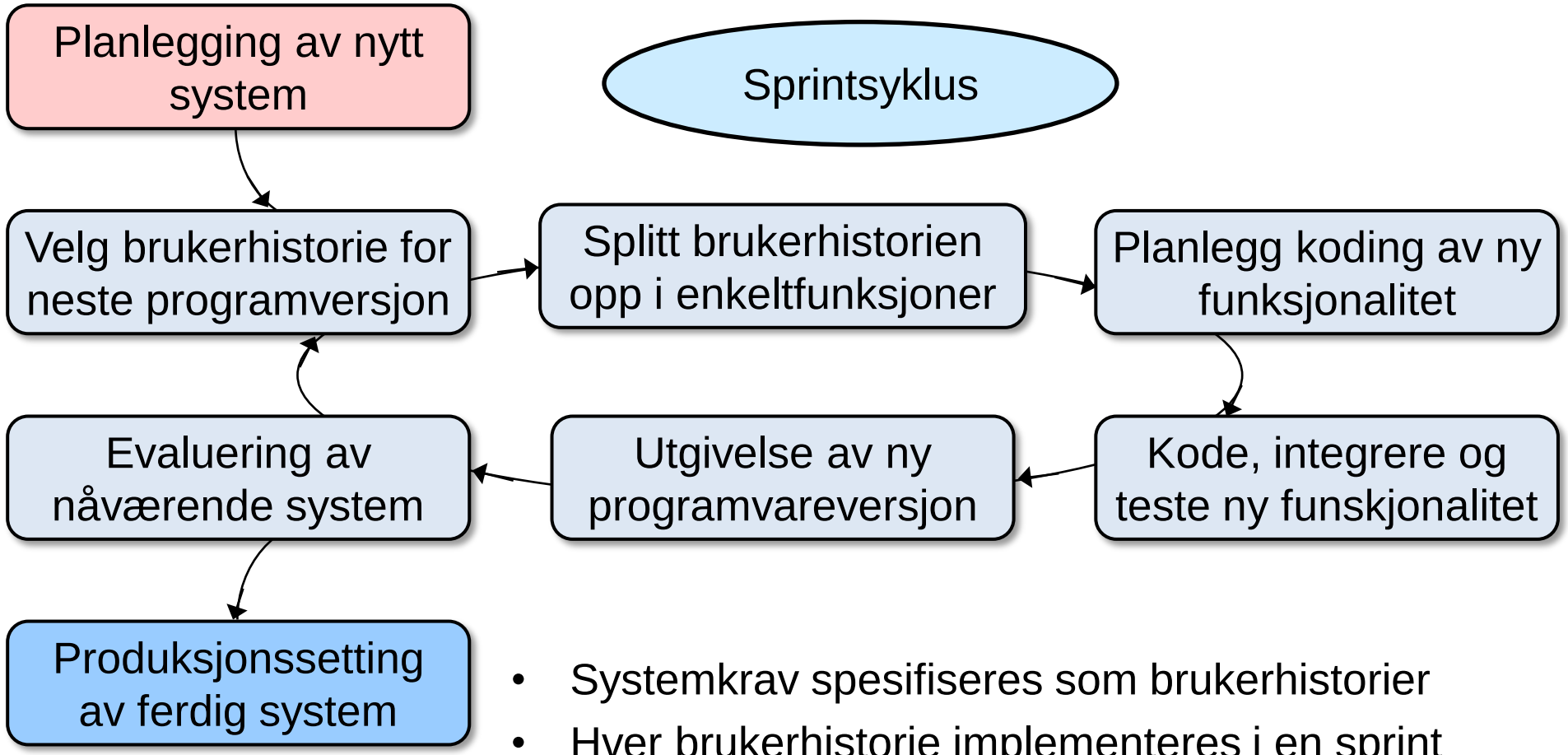
3. Sikker design

- Viktig del å spesifisere «sikre funksjoner», som er godt designet med hensyn til sikkerhet.
- Sikkerhetsfunksjoner som krypto, autentisering, logging etc er viktig. Designer må ha god kompetanse på disse.
- Trusselmodellering (mer om dette senere) er en viktig del her for å unngå sårbarheter

4. Sikker koding

- Målet med sikker koding er å unngå at sårbarheter bygges inn i systemet under koding og at sikkerhetsfunksjoner fungerer i henhold til krav og design
- Moderne utvikling benytter i stor grad tredjepartskomponenter, og det er viktig å forstå innvirkningen disse kan ha på sikkerheten. Usikre komponenter skal ikke brukes.
- Verktøy for kodeskanning og statisk analyse, gjennomgang av kode samt bruk av «sikre» programmeringsspråk er viktig her.

Smidig programvareutvikling



- Systemkrav spesifiseres som brukerhistorier
- Hver brukerhistorie implementeres i en sprint
- Fortsetter så lenge det er igjen brukerhistorier
- Systemet er ferdig når alle brukerhistorier er laget

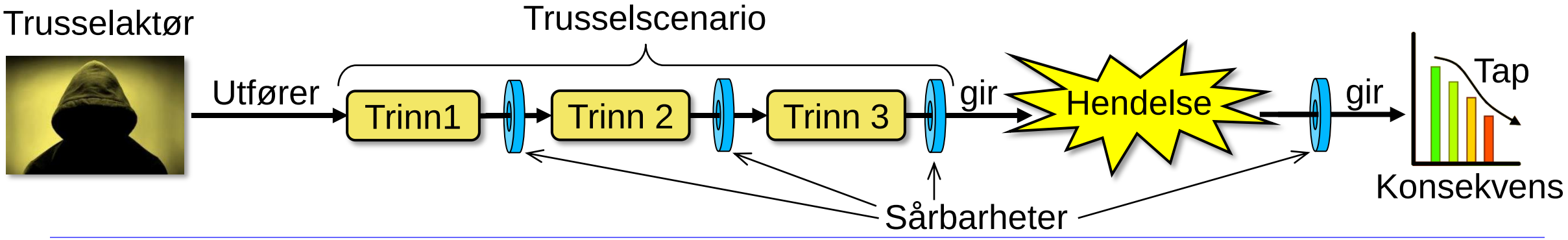
Sikker smidig programvareutvikling

- Sikker smidig programvareutvikling har tilleggsaktiviteter
  - … ved oppstart av prosjektet
  - … i hver sprintsyklus
  - … ved avsluttende evaluering av system
- «Sikker smidig» blir nødvendigvis litt mindre smidig

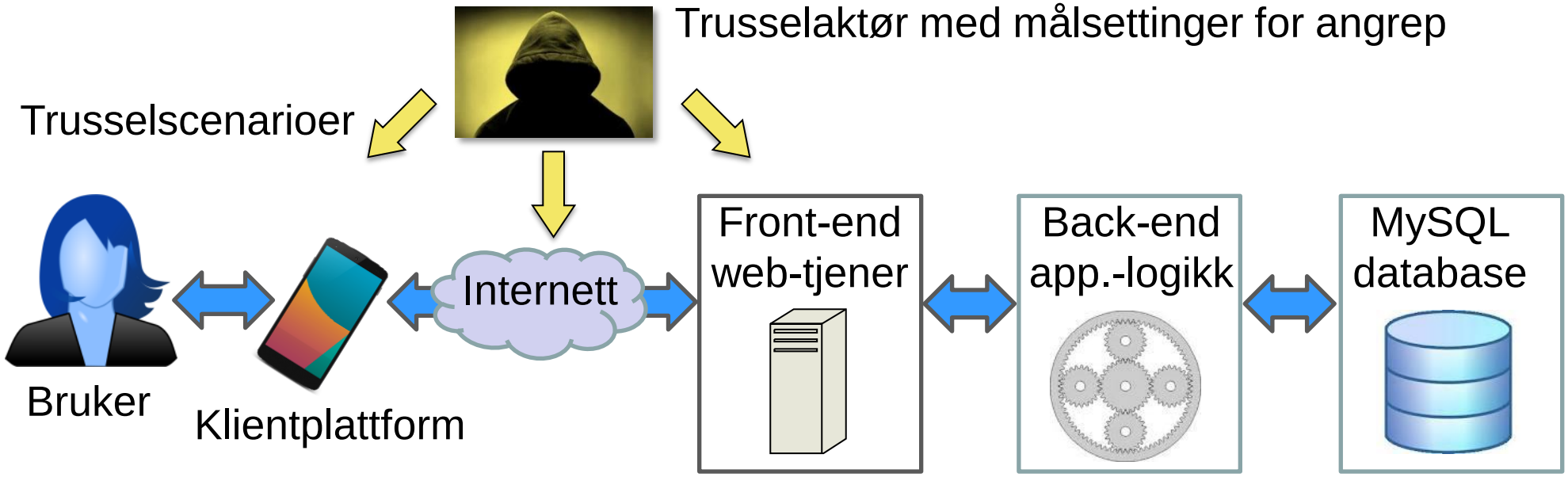
Verdier, Trusler, Sårbarheter og Tiltak

- **Verdier:** (Informasjons)ressurser som er av verdi for organisasjonen.
  - Data, systemer, applikasjoner, nettverk, enheter, tjenester, mennesker

    Mål for informasjonssikkerhet er å beskytte verdienes KIT, avhengig av behov.
  - Person(opplysnings)vern
- **Trussel:** Et potensielt angrepsscenario som kontrolleres av en trusselaktør, som kan skade organisasjonens verdier
- **Sårbarhet:** Mangel på sikkerhetstiltak mot trusler.
- **Sikkerhetstiltak** (Security Control): Metode for å forhindre trusler eller redusere konsekvenser



Trusselmodellering



- Trusselmodellering er å identifisere, analysere og beskrive relevante angrepsscenarier.
- I sikker smidig programvareutvikling skal trusselmodellering og en enkel risikovurdering utføres som del av hver sprint.
- Tenk: Hvordan kan denne nye funksjonen misbrukes eller angripes? Hvilke verdier kan bli skadet? Hvilke konsekvenser kan det få?
- Stopp eller reduser trusselen (fjern sårbarheter) under sprinten.

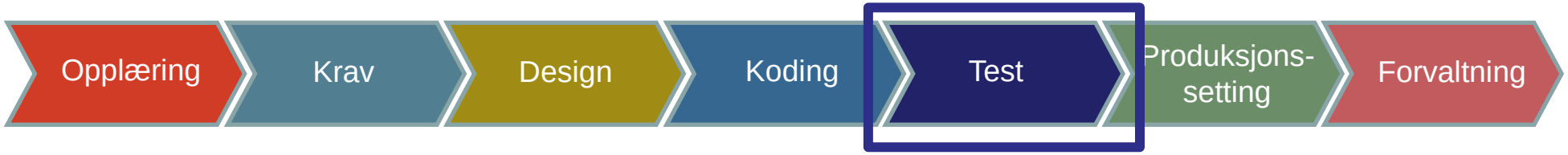
STRIDE Trusselmodellering for programvareutvikling

- Det finnes flere rammeverk for trusselmodellering.
- STRIDE er et populært rammeverk utviklet av Microsoft, hvor navnet kommer fra forbokstavene til 6 kategorier
- Hensikten er at disse kategoriene skal hjelpe å identifisere sårbarheter

Bokstav	Trusselkategori	Beskrivelse	Sikkerhetsbrudd
S	*Spoofing* Identitetstyveri	Kan en angriper få uautorisert tilgang ved å stjele en annens identitet?	Brudd på autentisitet
T	*Tampering* Tukling	Kan en angriper endre konfigurasjon eller data som prosesseres av systemet?	Brudd på integritet
R	*Repudiation* Benekting	Kan en angriper benekte misbruk fordi vi mangler spor og logger som peker ut angriperen?	Brudd på sporbarhet
I	*Information disclosure* Datatyveri og -lekkasje	Kan en angriper få tilgang til konfidensielle og personlige data?	Brudd på konfidensialitet
D	*Denial of service* Tjenestenekt	Kan en angriper blokkere eller minske tilgjengeligheten til systemet?	Brudd på tilgjengelighet
E	*Elevation of privilege* Utvidede tilganger	Kan en angriper oppnå utvidede tilganger og dermed bli en privilegert bruker?	Brudd på tilgangskontroll

Fase 5: Sikkerhetstesting

- Mål er å avdekke sårbarheter som ikke har blitt oppdaget i design- eller kodefasene.
- **Dynamisk testing/sårbarhetsanalyse** av den fullstendige programvaren sjekker funksjonalitet som blir synlig når alle komponentene er integrert sammen. Sjekker blant annet at bruker får tilgang til informasjon/funksjonalitet den skal (og ikke informasjon som bruker ikke skal ha tilgang til).
- **Penetrasjonstesting** går et steg videre og er et (autorisert) simulert angrep for å evaluere sikkerheten («etisk hacking» brukes ofte om dette). Skiller mellom
  - hvitbokstesting der angriper har informasjon om system på forhånd
  - svartbokstesting der angriper ikke har slik informasjon.
- **Fuzztesting** forsøker å fremprovosere feil i systemet ved å gi korrupte inputverdier (tilfeldig eller misformet data).



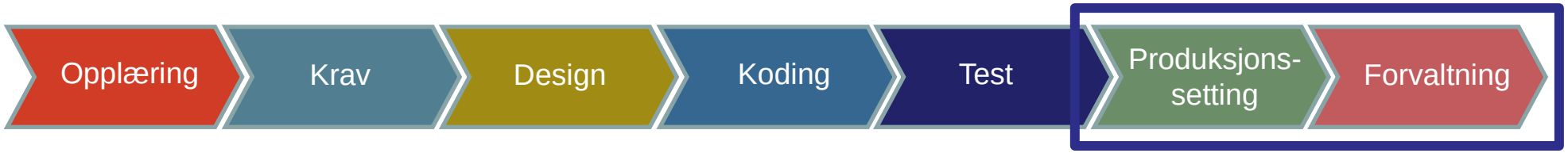
Fase 6: Produksjonssetting og Fase 7: Forvaltning

6. Produksjonssetting

- **Plan for drift, vedlikehold og hendelseshåndtering** må definere prosedyrer for drift (inkl. patching), avviksrapportering og hendelseshåndtering (mer om dette neste uke)
- **Formell godkjenning av produksjonssetting** vil kreve at det verifiseres og dokumenteres at alle krav til sikkerhet og personvern er oppfylt og identifiserte sårbarheter er tilstrekkelig fjernet. Formelt ansvar/mandat må defineres og relevant data og dokumentasjon arkiveres.

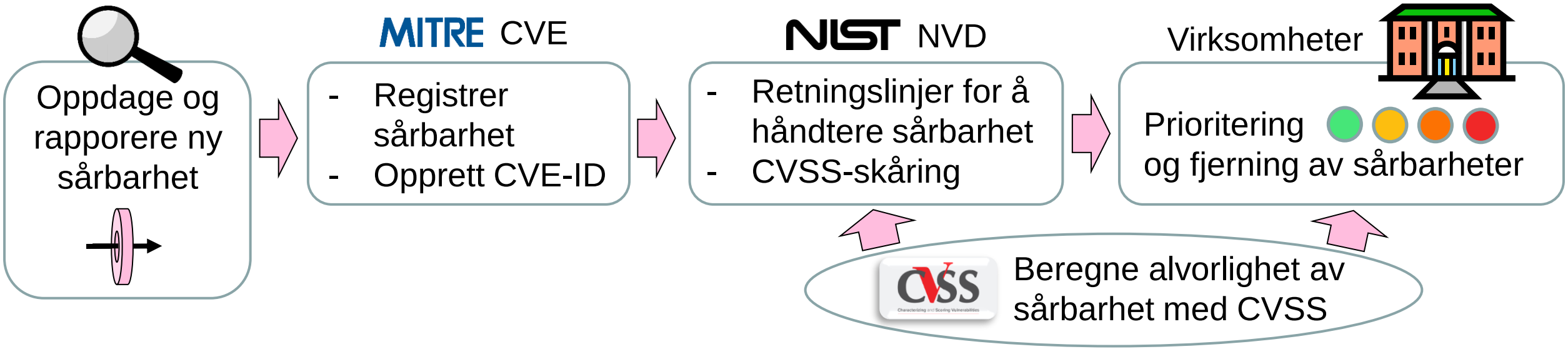
7. Forvaltning

- **Drift og vedlikehold** innebærer at prosedyrer og rutiner for drift og vedlikehold av programvare skal følges, også over tid. Revisjoner bør gjennomføres regelmessig og et ledelsessystem for informasjonssikkerhet bør være på plass. Man må klart definere hva som skal logges og hvordan loggene håndteres.
- **Avviks- og hendelseshåndtering** Avvik og hendelser skal rapporteres som beskrevet i planene. (Mer om hendelseshåndtering neste uke).



Fjerning av sårbarheter i programvare i produksjon

- I livssyklusen for håndtering av sårbarheter fins det flere trinn, fra tidspunktet da en sårbarhet oppdages og rapporteres, til den er fjernet i instanser av programvaren hos virksomheter.



CVE og CVSS

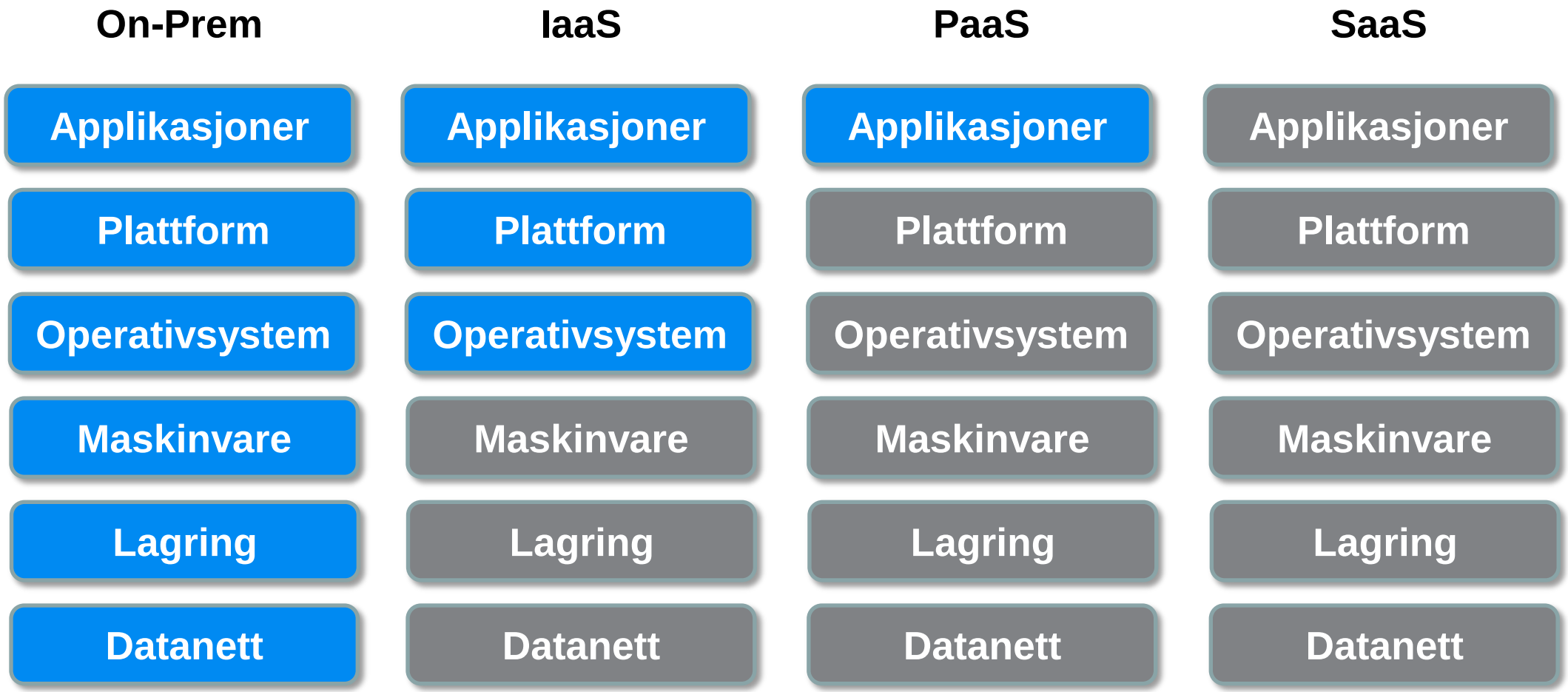
- CVE (Common Vulnerabilities and Exposures) er en database over sårbarheter i alle typer programvare.
- NVD (National Vulnerability Database) importerer data fra CVE, legger til retningslinjer for håndtering og fjerning av sårbarheten i den aktuelle programvaren, typisk med input fra programvareprodusenten.
- Alvorlighetsgraden beregnes med CVSS (Common Vulnerability Scoring System), basert på ulike faktorer, gir alvorlighetsskåring på en skala fra 0 til 10.

- Kritisk: 9,0 – 10
- Høy: 7,0 – 8,9
- Middels: 4,0 – 6,9
- Lav: 0,1 – 3,9

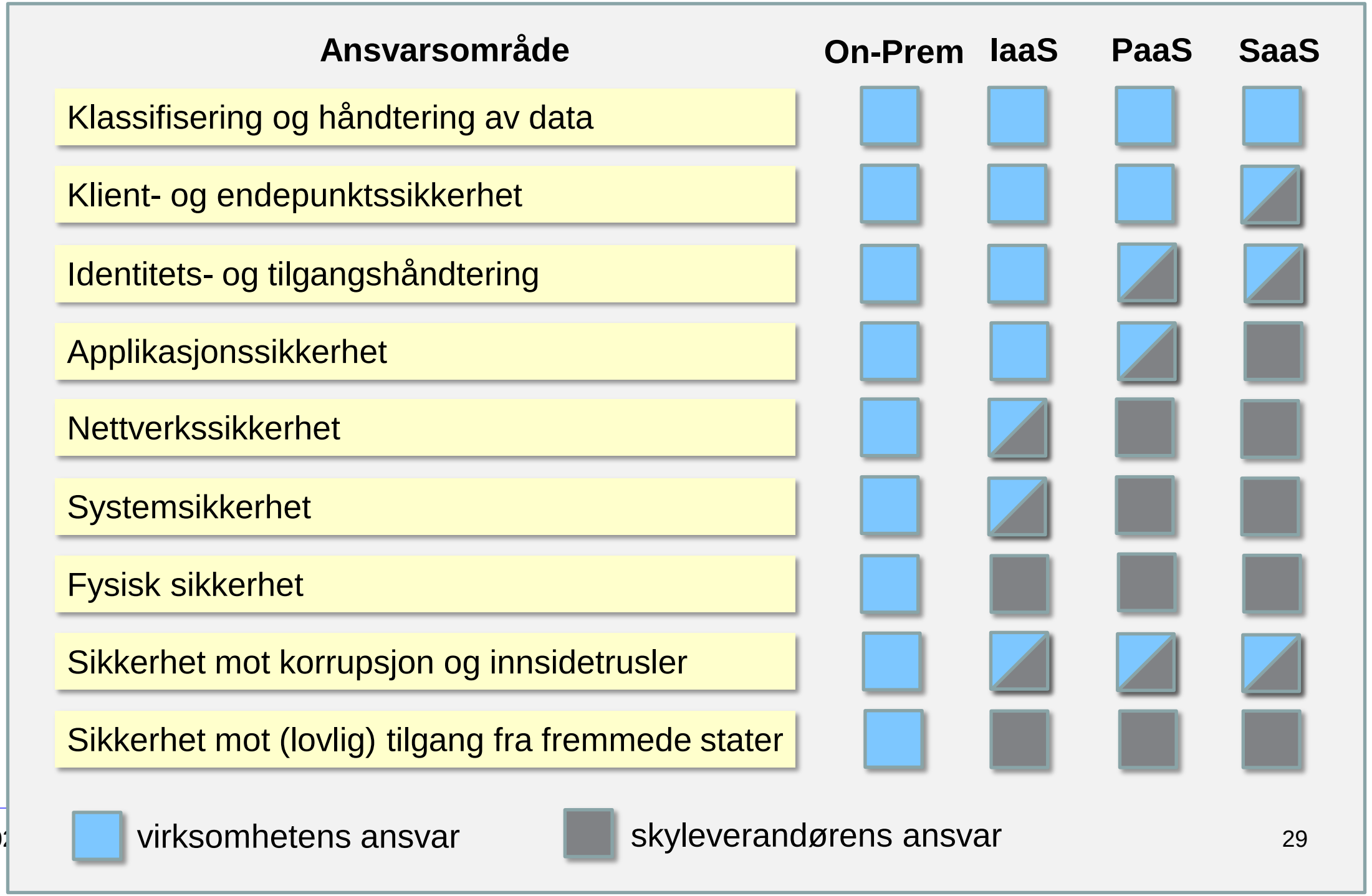
SBOM: Digitale stykklister

- SBOM (Software Bill of Materials) betyr at det lages «digitale stykklister» over alle komponenter som inngår i systemer eller programvareapplikasjoner.
- Inkluderer alle programvarebiblioteker og eksterne komponenter som lenkes inn.
- Gjør det mulig å kartlegge alle kjente sårbarheter i et system eller applikasjon.
- SBOM-initiativet startet i 2018, og er etterhvert blitt en viktig del av programvaresikkerhet og risikostyring for leveransekjeden.
- SBOM er et samarbeid mellom ulike aktører, styrt av CISA (Cybersecurity and Infrastructure Security Agency) i USA.

Delt ansvar ved ulike skymodeller



Delt ansvar for sikkerhet ved ulike skymodeller



Sikkerhet i skyen

- Skyleverandører har ofte betydelig kompetanse og ressurser for å opprettholde høy grad av sikkerhet i sine infrastrukturer, f.eks. ved rask sikkerhetsoppdatering, effektiv sikkerhetsmonitorering og deteksjon, og hendelsesrespons.
- Aspekter som kan true sikkerhet ved bruk av skyløsninger er f.eks.:
  - Identitets- og tilgangshåndtering ved bruk av skytjenester kan utgjøre en sårbarhet hvis den er dårlig implementert. Skytjenester er teknisk sett tilgjengelig fra hele Internett, slik at det trengs robuste løsninger for beskyttelse av tilgang. *«Identity is the new security perimeter»*
  - Potensiell korrupsjon eller utro tjenere hos skyleverandøren. Sannsynlighet for dette er vanligvis svært lav, men kan øke i land der nivået av korrupsjon er relativt betydelig og rettsikkerheten er relativt lav.
  - Lovlig tilgang til virksomhetens data fra myndigheter i land der skyleverandøren er lokalisert. Dette kan skje som del av etterretning eller etterforskning av kriminalitet, uten at virksomheten nødvendigvis blir informert.

Slutt på presentasjonen