

# FRA SPAGHETTI-INTEGRATION TIL ORDEN I EGET HUS

White Paper

# INDHOLDSFORTEGNELSE

<b>Business integration</b> .....	<b>4</b>
Den ideelle verden .....	4
Integrationsbølger historisk .....	4
Første bølge. <i>In house integration</i> .....	4
Anden bølge. <i>Integration til web</i> .....	4
Tredje bølge. <i>"Alle til alle" integration</i> .....	4
<b>Faldgruber ved ad hoc integration</b> .....	<b>5</b>
Kompleksitet .....	5
Problemer med dokumentation .....	6
Mangel på overblik og klar besked.....	6
Svære størrelser .....	6
<b>Et godt værktøj</b> .....	<b>7</b>
Krav 1. Overblik .....	7
Krav 2. Skal kunne konfigureres .....	7
Krav 3. Åbent for udvidelser og specialtilpasninger .....	7
Krav 4. Dokumentation .....	7
Krav 5. Anerkendte standarder.....	7
Krav 6. Klar besked om fejl og status .....	7
<b>Seks anbefalinger</b> .....	<b>8</b>
Anbefaling 1. Start i det små .....	8
Anbefaling 2. Trin for trin .....	8
Anbefaling 3. Cost vs. benefit.....	8
Anbefaling 4. Mød systemerne, hvor de er .....	8
Anbefaling 5. Kend og dokumentér din løsning .....	9
Anbefaling 6. Vælg ét internt format.....	9
<b>Om Bording Data</b> .....	<b>9</b>

# BUSINESS INTEGRATION

Disciplinen business integration tilsigter at automatisere kommunikationen mellem leddene i virksomhedens værdikæde. Der er store effektiviseringsgevinster ved en sådan automatisering, men også mange faldgruber. Ofte opstår integrationer som ad hoc løsninger, som får lov at vokse sig større, indtil kompleksiteten bliver en klods om benet.

Dette indlæg gennemgår business integrationens fordele og faldgruber, de historiske bølger samt anbefaler hvilke krav man bør stille til et nyt integrationssystem. Afslutningsvis gennemgås seks anbefalinger den, der påtænker at implementere automatiseret kommunikation eller omlægge eksisterende.

Der kan være mange grunde til at ønske en automatiseret integration mellem de it-systemer, som understøtter virksomhedens værdikæde. Det kan dreje sig om kontakten til fysisk separate lokationer, partnere, leverandører og kunder. Er man afhængig af manuelle arbejdsgange i informationsudvekslingen, vil der uvægerligt opstå menneskelige fejl, være en større forsinkelse samt en ofte unødigt replikering af data.

Manuelle processer er tidsrøvere og dermed dyre. De er ofte en flaskehals i forhold til lidt større datamængder og omstilling til kommunikation med nye parter. Alle disse forhold trækker i den forkerte retning i en verden, hvor rationalisering og omstillingsevne er afgørende for overlevelse.

## DEN IDEELLE VERDEN

I den ideelle verden er al informationsudveksling og -behandling, der ikke kræver menneskelig interaktion, automatiseret. Alle systemer i værdikæden er sammenbundet i et elektronisk nervesystem, hvis formål er at reagere prompte på ændringer i indre og ydre forhold. Virksomhedens drift understøttes straks med den seneste status, og systemet tilgodeser hurtig, enkel og billig indkobling af nye systemer i den elektroniske værdikæde. Bliver leverandør B pludselig 10 % billigere end leverandør A i en periode, så skal det være muligt at udnytte denne fordel, inden den forsvinder igen.

## INTEGRATIONSBØLGER HISTORISK

Integration af it-systemer i netværk blev tidligt populært. Man kan tale om tre overordnede integrationsbølger, som har ramt verden i et forsøg på at høste fordelene ved automatisering.

### Første bølge. In house integration

Den tidligste form for integration startede inde i virksomheden og bevægede sig herfra ud i værdikæden. Man

fokuserede på at få virksomhedens interne program til at tale sammen, netop fordi den manuelle replikering af data mellem systemerne var tung. Regnskab skulle udveksle data med indkøb, som skulle udveksle data med lagersystemet osv. Det var naturligt at begynde med det nære, blandt andet fordi teknologien på internettet stadig var ny. En sådan applikationsintegration er til stadighed aktuel i mange virksomheder.

### Anden bølge. Integration til web

Med internettets fremmarch blev der mulighed for at tiltrække kunder via nettet. I mange brancher blev det en vigtig konkurrenceparameter at kunne tilbyde nethandel. Denne integration betød døden for de mange spillere, som levede af vareformidling. Nye virksomheder skød op – man behøver blot at nævne Amazon eller eBay.

### Tredje bølge. „Alle til alle“ integration

Da virksomhedens berettigelse var tilgodeset, var det naturligt at spejle efter nye gevinster i baglandet i udvekslingen med de øvrige led. Begrebet Business Integration voksede frem og betegner den automatisering af værdikæden, som typisk foretages mellem virksomheder eller institutioner, der udgør afgrænsede selvstændige enheder. Der kan være tale om partnere, leverandører, kunder, søsterselskaber eller andre, der bidrager med informationer og hændelser til virksomhedsprocessen. Dette begreb omfatter således hospitaler, der ønsker at automatisere journaludveksling, biblioteker, der ønsker at fremstå som samlet lånesystem, og den almindelige virksomhed, der ønsker at erstatte papirbaserede varekataloger, ordrer, fakturaer med online services.

Integrationsbestrebelse er en stadig udfordring i den åbne verden, vi lever i. Gamle it-systemer udskiftes med nye, enkelte led i værdikæden udskiftes til fordel for nye partnere osv. Integration er en disciplin, der er kommet

for at blive, og man bør som virksomhed forholde sig til dette i form af en veldefineret overordnet strategi. Mange virksomheder har historisk reageret på udfordringen på ad hoc basis. Den manglende strategi har ikke nødvendigvis været et resultat af mangel på kompetence, men skal snarere ses i lyset af, at området er nyt samt det forhold, at man i praksis ofte ikke har kunnet vente

på formuleringen af en overordnet politik, når man stod med et konkret behov for dataudveksling, som kunne give umiddelbare besparelser. I den forstand minder ad hoc integration om ad hoc programmering i systemudviklingens barndom, hvor metoder og best practices var teoretiske idealer i deres vorden, hvorimod konkrete implementeringer mindede om en italiensk pastaret.

## FALDGRUBER VED AD HOC INTEGRATION

Der er klare økonomiske fordele ved at automatisere dataudvekslingen. Derfor er mange systemer blevet bundet sammen, når et behov er opstået. Én it-kyndig i organisationen har en sen natte-time besat af en fremragende ide, som mindst inkluderede fire teknologier og et par små ændringer i ordresystemets kodebase fået strikket en kombination af scripts sammen, som formår at udtrække, konvertere og/eller eksportere data til et partnersystem eller omvendt konvertere og importere til egne systemer. Måske har han/hun endda allieret sig med en ligesindet hos udvekslingspartneren. Chefen har været tilfreds, fordi løsningen indebar, at tingene gled lettere og har måske ikke interesseret sig meget for præmisserne for løsningen.

### KOMPLEKSITET

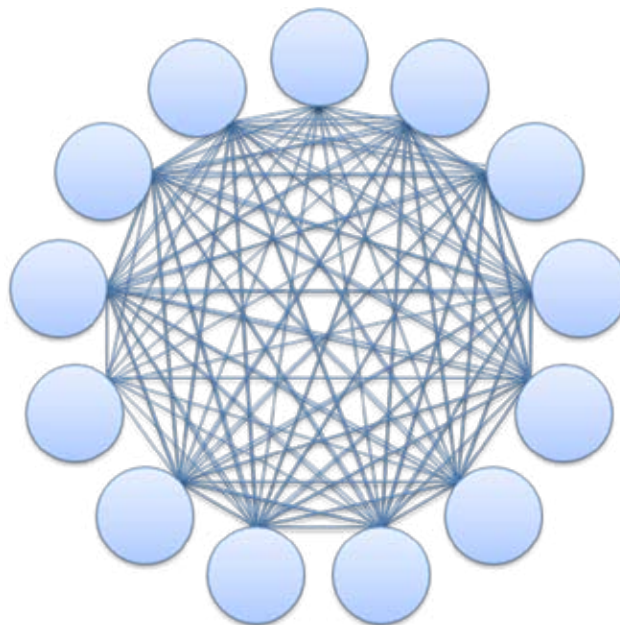
Uden overordnet koordination vil de mange velmente tiltag skabe et system af forbindelser udviklet i mange forskellige teknologier (unix/dos-scripts, basic, Java, c-kode, SQL osv.). Noget vil eksistere som eksterne komponenter i forhold til systemerne, der integreres, men ofte ser man også specialtilretninger i baggrundssystemerne. Det samlede kompleks vil typisk være en blanding af integration, konvertering, transformation og kommunikation. Der er sjældent gjort noget for at sikre klare grænseflader mellem de logiske lag i løsningen.

God integration er en disciplin med egne værktøjer i form af teknologier, standarder og best practices. Disse vil der ofte ikke være kendskab til hos en enkelt udvikler eller supporter, som beskæftiger sig med andet i hverdagen. At reducere mapningskompleksitet ved brug af et fælles format er en kongstanke i mange-til-mange integration, men ikke et princip der gives grund til at reflektere over, når integrationen kommer gradvist.

Hér rammes man af en række komplekse problemstillinger; jo flere teknologier, der er i spil og des dårligere den indre arkitektur er, des større specialviden vil det kræve at forstå og vedligeholde en løsning.

### PROBLEMER MED DOKUMENTATION

Den almindelige træghed med at udarbejde og vedligeholde dokumentation af programmel gælder også, når vi taler integration. Det er ekstra uheldigt, da heterogene, distribuerede systemer ofte er så meget mere komplekse at overskue end selv store stand-alone-programmer. Dette skyldes deres distribuerede natur, da den logiske funktionalitet og de indgående løsningsfragmenter er spredt fysisk, og der er ikke ét program eller bibliotek, man kan inspicere. Hertil kommer at forskellige formater med forskellig semantik indgår. En ordre fra en partner ser måske ikke helt ud, som ordrer i det interne system, så en hel del viden om konvertering og fortolkning af felter i ordren bliver indlejret i løsningen.



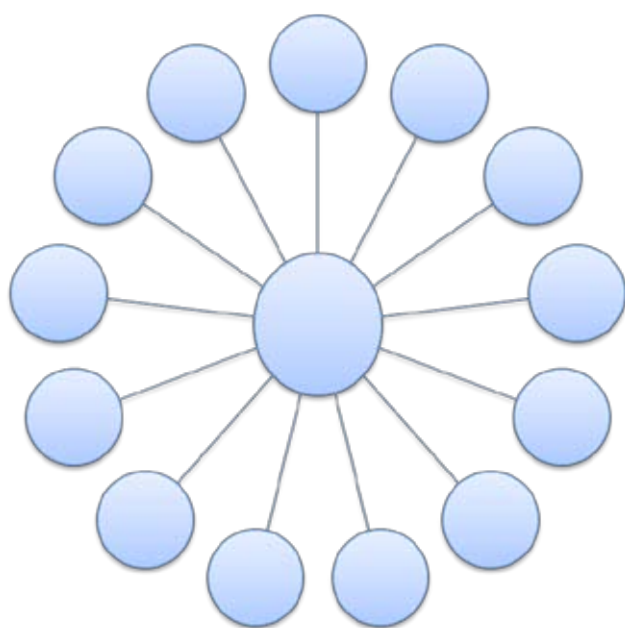
Figur 1: Punkt-til-punkt løsning

### MANGEL PÅ OVERBLIK OG KLAR BESKED

En ad hoc integrationsløsning giver typisk ikke noget samlet overblik over de driftsmæssige forhold, selvom den er godt dokumenteret. Fik vi morgens lagerliste?

Kører ordreimporten, som den skal? Når flere systemer kommunikerer over netværk, er fejlkilderne mange og behovet stort.

I bedste fald har udvikleren måske sørget for at skrive fejl fra programmerne i en log-fil. Når dataudvekslingen fungerer fejlfrit, er dette et mindre problem, skønt man ikke vil vide præcist, hvad og hvor meget, der er udvekslet, ikke mindst hvornår. Det kan give problemer, hvis en partner insisterer på at have sendt et elektronisk dokument, som man ikke kan finde i sit eget system og vice versa.



Figur 2: Centralt integrationsværktøj

Værre er det, når ting fejler. Der skrives i en log, men hvilken log? Jo flere systemer der er involveret med en eller flere logfiler på forskellige maskiner desto værre. I praksis kan sådanne fejl ligge upåagtet hen i dage eller uger, og måske reageres der først, når et alvorligt forretningsmæssigt problem opstår. Endnu værre er det naturligvis i de tilfælde, hvor fejl ikke håndteres af systemerne, der kører uanfægtet videre.

## SVÆRE STØRRELSER

Distribuerede systemer er i sig selv svære størrelser, men at integrere dem via ugenomsigtige løsninger, komplicerer tingene yderligere. I stedet er der behov for **transparens og et overblik**.

Den tekniske kompleksitet og ringe dokumentation kombineret med et naturligt tab af viden, når en medarbejder rejser, gør, at ingen har lyst til at pille ved løsningen. Ofte er det en alment accepteret holdning til sådanne systemer, at det er bedre at lade dem være. De, der ender med at skulle ned i materien, vil være udviklere med specialviden, og de vil skulle bruge meget tid. Med andre ord bliver vedligeholdelse og udbygning meget dyr. Det hindrer let og hurtig opkobling mod nye led i værdikæden, der kunne give forretningsmæssige fordele, som i stedet bliver ædt op af udgifter (hvis man da overhovedet har mod på at komme i gang).

Der er opsamlet stor erfaring med integration i de seneste årtier, og der er på den baggrund opstået en vis konsensus om, hvordan gode og dårlige løsninger ser ud. Derudover har de professionelle integrationsværktøjer nået en modenhed og et prisniveau, der gør dem attraktive også for mindre virksomheder.

# ET GODT VÆRKTØJ

Her er seks krav til et nyanskaffet integrationsværktøj. Valget af et godt værktøj er det første skridt på vejen til at få styr på sin virksomhedsintegration.

## KRAV 1. OVERBLIK

Professionelle integrationsværktøjer skal give et centralt overblik over den samlede integration. Det er en stor fordel, hvis integrationsværktøjet også giver et grafisk visuelt overblik af transaktionerne både på overordnet og på detailniveau. Det gør den centrale desktop lettere for nye såvel som gamle udviklere at vedligeholde løsningen; man kan se hvilke samarbejdspartnere, der udveksles med, hvilke servere, der indgår, og hvordan dokumenter transformeres - typisk i en række trin. Transparens giver kort sagt tryghed.

## KRAV 2. SKAL KUNNE KONFIGURERES

Selve arbejdet med integrationsløsningen bør så vidt muligt foregå ved konfiguration og definition, da man helst skal løfte sig et stykke op over en egentlig programmering af integrationen. Det er muligt, da integrationsteknologier baserer sig på velkendte, formaliserede protokoller og standarder. Fordelen er, at man sparer tid, og der opstår færre fejl, jo mindre der skal programmeres. Integrationsværktøjer skal også tillade, at man konfigurerer kommunikationen med omverdenen ved at angive basale data til komponenter, som går under navne som **adaptors**, **connectors** o.l. Eksempelvis vil man skulle angive simple værdier som ip-nummer, folder, filnavne og login-information for at hente filer automatisk fra en FTP-server.

Integrationsværktøjer skal også tilbyde konvertering mellem forskellige dokumentformater, og denne konvertering skal kunne foretages ved angivelse af formatets type: XML, komma-separeret tekst og hvordan felter fra input mappes til output.

## KRAV 3. ÅBENT FOR UDVIDELSER OG SPECIALTILPASNINGER

Det er vigtigt, at værktøjet er åbent for udvidelser. I det omfang behovene er meget specielle, som hvis man fx har brugt et internt proprietært dataformat, skal et integrationsværktøj have ét (eller flere) sprog, der tillader, at man skriver egen databehandling samt udvidelsesmuligheder, der typisk tillader plug-in af egne I/O-komponenter o.l.

## KRAV 4. DOKUMENTATION

Der skal være adgang til viden om brug af værktøjet, og dokumentationen skal være i orden. Manglende dokumentation kan i praksis forsinke og fordyre udvikling betydeligt, men også gøre oplæring af nye medarbejdere vanskelig. Der skal foreligge brugbar dokumentation med eksempler, der gør det let at komme i gang, samt adgang til konsulentbistand, hvis man har behov for dette. Derfor er det en god idé at undersøge værktøjets dokumentation, inden man anskaffer sig det.

## KRAV 5. ANERKENDTE STANDARDER

Forskellige værktøjer løser implementering af integrationen forskelligt og bruger forskellige begrebsapparater i arbejdet med dem. Det er en stor fordel, hvis værktøjet anvender standarder fx i forbindelse med transformation af data, interaktion med baggrundssystemer o.l. Eftersom XML ofte vil være dataformatet, der arbejdes med i dele af processen, bør teknologier omkring XML (XSD, XSL, XQuery osv.) være understøttet til brug for validering, transformation osv. Man vil som udvikler kunne finde en mængde dokumentation og eksempler fra andre kilder, så indlæring, udvikling og overdragelse bliver lettere.

## KRAV 6. KLAR BESKED OM FEJL OG STATUS

Lige så vigtigt, som det er at have et klart billede af løsningens anatomi, er det at have et klart billede af den daglige dataudveksling. Et godt integrationsværktøj skal tilbyde et centralt overblik over, hvad der er udvekslet og med hvilken status. Det kan være lige så værdifuldt at vide, at noget er gået **godt**, som at noget er gået **galt**. Samfundet bliver til stadighed mere afhængigt af automatiserede systemer og manglende opmærksom på en fejl kan betyde alt fra sagsanlæg til, at en skoleklasse ikke får sin mælk leveret. Der må derfor kunne reageres prompte, når der uundgåeligt opstår fejl fx i netværkskommunikation eller ved modtagelse af ugyldige dataformater.

Gode systemer er fejltolerante. Det betyder bla. at de tager højde for, at andre delsystemer ikke fungerer altid og derfor formår at udskyde dataudveksling til et senere tidspunkt, hvis der aktuelt ikke er kontakt. Herved kan temporære problemer løses automatisk uden menneskelig indblanding. Andre typer af fejl kræver derimod menneskelig interaktion. Her skal integrationsproduktet straks kunne advisere fx en administrator via e-mail eller SMS.

# SEKS ANBEFALINGER

Her er seks anbefalinger til, hvordan man kommer i gang med at frigøre sig fra sin „spaghetti integration“ og lægge en langsigtet integrationsstrategi for virksomheden.

## ANBEFALING 1. START I DET SMÅ

Det kan være en stor øjenåbner at udvælge nogle enkelte scenarier og få implementeret integration for disse som et **proof-of-concept**. Herved får man erfaringer med værktøjets muligheder og begrænsninger samt hele integrationsprojektets udfordringer og potentiale. Ofte vil det give nye ideer til automatisering af processer i virksomheden.

## ANBEFALING 2. TRIN FOR TRIN

Migrer trin for trin. En gradvise migrering har en række fordele; risikoen for, at projektet løber af sporet er mindre, da graden af kompleksitet vil være mindre. Derudover bliver man klogere på systemet og dets muligheder for hver fase. Læringskurven er stejl i starten, men flader ud jo flere del-projektet man kommer igennem.

Ud over at gøre opgaven mere overskuelig og hurtigere at tage i brug, er risikoen mindre, fordi man højst kan rammes af problemer i den delmængde, man har migreret. Skulle det værste ske, kan man fokusere på at udbedre problemer i denne delmængde.

Der kan dog være tungtvejende grunde til, at man ikke kan anlægge denne strategi. I sådanne tilfælde kan man inddеле integrationerne i et antal faser, og trin for trin konstatere, hvor migreringen har været en succes.

## ANBEFALING 3. COST VS. BENEFIT

Vær bevidst om, hvor den største fordel opnås med den mindste indsats. Optimeringen af de sidste 20 % udgør måske 80 % af den samlede omkostning samt, at optimeringen af de sværeste 20 % kan give mere gevinst end resten tilsammen. Denne „naturlov“ gælder også for automatisering af processer. Begynd derfor, hvor afkastet er størst. Her vil omfanget af manuel interaktion og volumen i udveksling være betydende. Manuelle processer er oplagte kandidater, idet simpel udveksling og behandling er det, som IT udfører bedre end mennesker, og ressourcer kan frigives til andet arbejde. Dog vil det næppe give mening at automatisere en proces, der tager en enkelt mandetime hvert halve år, selvom det evt. ville være muligt.

## ANBEFALING 4. MØD SYSTEMERNE, HVOR DE ER

Tag en pragmatisk tilgang til automatisering. Eksisterende

de systemer vil typisk have en række kanaler ad hvilke, de kan tilgås og som kan udnyttes direkte til at sammensætte en automatiseret udveksling mellem parterne. Fx understøtter et system måske import/eksport af data via filer i filsystemet eller afsending af e-mails med den information, der skal udveksles. Kombineret med en lokal FTP- eller mail-server har man herved tilstrækkelig adgang til omverdenen til, at integrationsværktøjet kan forestå levering og konvertering til omverdenen ad de kendte kanaler.

Denne form for integration er **non-intrusiv**, da de eksisterende systemer ikke skal modificeres. Det er ikke en god idé at udbygge og modificere hver eksisterende applikation, der skal indgå i integrationsløsningen. Det skyldes, at det kræver special viden, og at det er komplekst og dyrt.

Fordelen ved at lade integrationen være „ekstern“ er, at den kan foregå i et centralt værktøj. Det kan gøres hurtigere og vedligeholdes langt lettere. Nye partnere kan lettere kobles på det eksisterende setup og central konverteringslogik kan ofte genbruges.

I de tilfælde, hvor der skal reeltids-integreres, fordi en bruger ønsker øjeblikkelig status på en transaktion, kan man sikre online synkron integration til back-end-systemerne, men i mange tilfælde er det ikke nødvendigt, og derved kan der skabes en ligeså velfungerende, men langt billigere løsning uden.

Hastighed og sikkerhed kan ofte være af stor betydning. Som oftest er det underordnet, om en ordre formidles på 10 sekunder eller 2 minutter, hvis udgangspunktet var manuelle arbejdsgange. Det væsentlige er, at man har automatiseret processen, og at dens data flyder mellem systemer i løbet af minutter.

Med hensyn til sikkerhed og pålidelighed kommer mange eksisterende transportprotokoller (secure FTP, HTTPS osv) med en snes år på bagen og samme certifikatbaserede sikkerhed på niveau med nye teknologier som webservices. De udgør dermed en hensigtsmæssig og næsten allestedsnævrende potentiel infrastruktur og er skabt til formålet.

## ANBEFALING 5. KEND OG DOKUMENTÉR DIN LØSNING

Man skal sørge for at opbygge og vedligeholde en systembeskrivelse f.eks. ved brug af en wiki, som omfatter eksterne parter, udvekslinger med disse, konver-

teringer og transformationer (hvordan, hvorfor) samt arbejdsgange ved problemer.

Mange parter, formater og fysiske maskiner er i spil, og en integrationsarkitektur med komplekse konverteringer og databerigelser kan være nødvendig. Driften af et sådan system vil have arbejdsgange, som skal kendes og kunne anvendes hurtigt i tilfælde af fejl.

## **ANBEFALING 6. VÆLG ÉT INTERNT FORMAT**

Lad eksterne formater forblive eksterne. Tit vil mange dokumenttyper (ordrer, fakturaer, kvitteringer, lagerlister osv.) være i spil. Dertil kommer, at forskellige parter ofte vil bruge forskellige repræsentationer (EDI, XML, CSV, SOAP osv.) måske baseret på forskellige standarder, ebXML, EDI, OIO. Dvs. forsøge at „normalisere“ de forskellige repræsentationer til én „universel“ intern repræ-

sentation (aka in-house format) – fx at have ét ordreformat, ét fakturaformat osv. og så konvertere de eksterne formater til denne repræsentation. De-facto standard er i dag at bruge XML som internt format, og der er ikke noget oplagt alternativ. Det interne format må indeholde alle de data, som er relevant på tværs af partnere, og det vil typisk være nødvendigt at berige datastrømmen i de tilfælde, hvor bestemte værdier ikke var til stede i det oprindelige format.

Fordelen ved denne fremgangsmåde er, at al databehandling, som vil være nødvendig på tværs af partnere, kun skal skrives én gang – nemlig for in-house-formatet. Generelt skal man undgå at replikere funktionalitet, blot fordi forskellige formater er i spil. Herved øges løsningens gennemsigtighed, og vedligeholdelsen forenkles væsentligt.

### **Om Bording Data A/S**

Bording Data A/S er officiel Microsoft Gold Partner og har over de seneste 10 år udviklet og leveret integrationsproduktet AGETOR® til en række mellemstore og større virksomheder i ind- og udland. I skrivende stund medvirker AGETOR® til applikations- og business integration i mere end 100 installationer, blandt hvilke kan nævnes JP/Politikens Hus, Øresundsbron, BAUHAUS, DSV Miljø, Elbodan, hummel og Super Best, Malmø Stad, Stockholms Universitet m.fl.

I Bording Data A/S tilstræber vi at levere hele løsninger i samarbejde med vore kunder og deltager gerne i alle faser af projekterne. For mere information kontakt Bording Data på +45 43 24 54 24, salg@bording.dk eller besøg [www.bordingdata.dk](http://www.bordingdata.dk)

[www.bordingdata.dk](http://www.bordingdata.dk)

Bording Data A/S

Baldersbækvej 24-26  
DK-2635 Ishøj

Bredskifte Allé 11  
DK-8210 Århus V.

[kontakt@bording.dk](mailto:kontakt@bording.dk)

+45 43 24 54 24