

ORCHESTRA
TEAM



Integrating Drupal to Business Platforms

Considerations, Techniques, and Opportunities

Chris Strahl



Brad Windecker



What are we covering?



1. Overview of ERP
2. Traditional Integration Model
3. Platform Integration Model
4. Power of a CMS Platform
5. Challenges to Overcome
6. Integration Methodologies
7. Q & A

ERP? I Can't Even Spell That!



Enterprise Resource Planning

The initials **ERP** originated as an extension of MRP (Material Requirements Planning and Manufacturing Resource Planning^[3]; later manufacturing resource planning) and CIM (Computer Integrated Manufacturing). It was introduced by research and analysis firm Gartner in 1990. **ERP** systems now attempt to cover all core functions of an enterprise, regardless of the organization's business or charter. These systems can now be found in non-manufacturing businesses, non-profit organizations and governments.^[4]

To be considered an ERP system, a software package must provide the function of at least two systems. For example, a software package that provides both payroll and accounting functions could technically be considered an ERP software package.^[citation needed]

- Thank you Wikipedia

The simplified real-world term:

Business Management and
Accounting System

How do Businesses Use ERP?



- Business Management & Accounting
 - Accounting / Financials
 - Inventory / Logistics / Manufacturing
 - Sales
 - Purchasing
 - CRM
 - Reporting / Planning / Budgeting
- Platform technology
 - Modular architecture
 - Extensible (usually an API)
- Typically they have a **single database**
- Typically they **do not** directly service customers or external users

Who builds ERP Systems?



Who uses ERP Systems?



Large Global Enterprise

- \$500M+ Revenue & 1,000+ Employees
- Large # of systems, processes, applications



Midsize Companies

- \$50M - \$500M in Revenue & 100 - 999 Employees
- Multiple Systems, Multi-location, complex IT

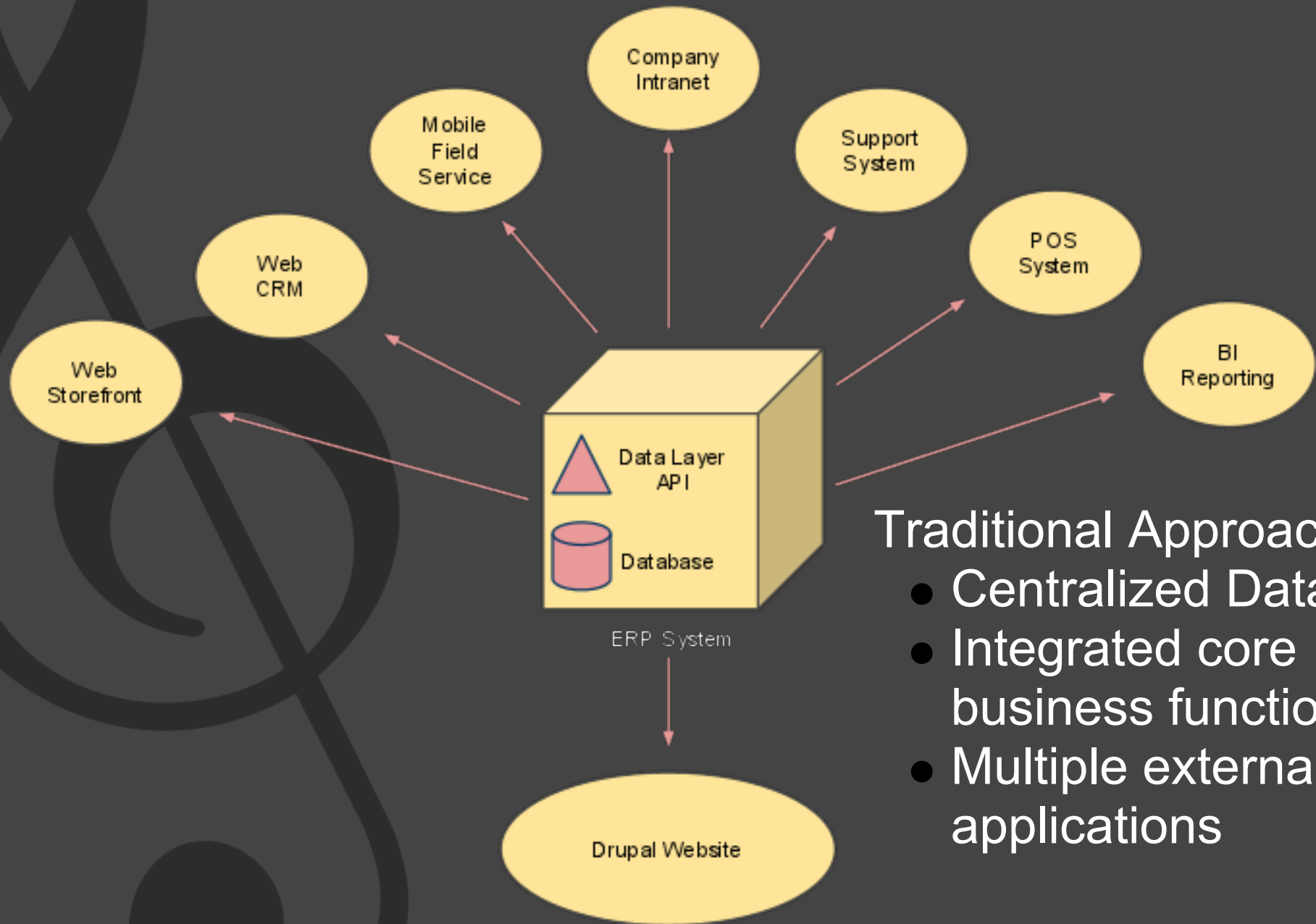


Small Business

- Up to \$50M in Revenue & 1-100 Employees
- Need for simple integration of ERP to web, often no in-house IT



The Traditional Setup for a Small Enterprise



Traditional Approach:

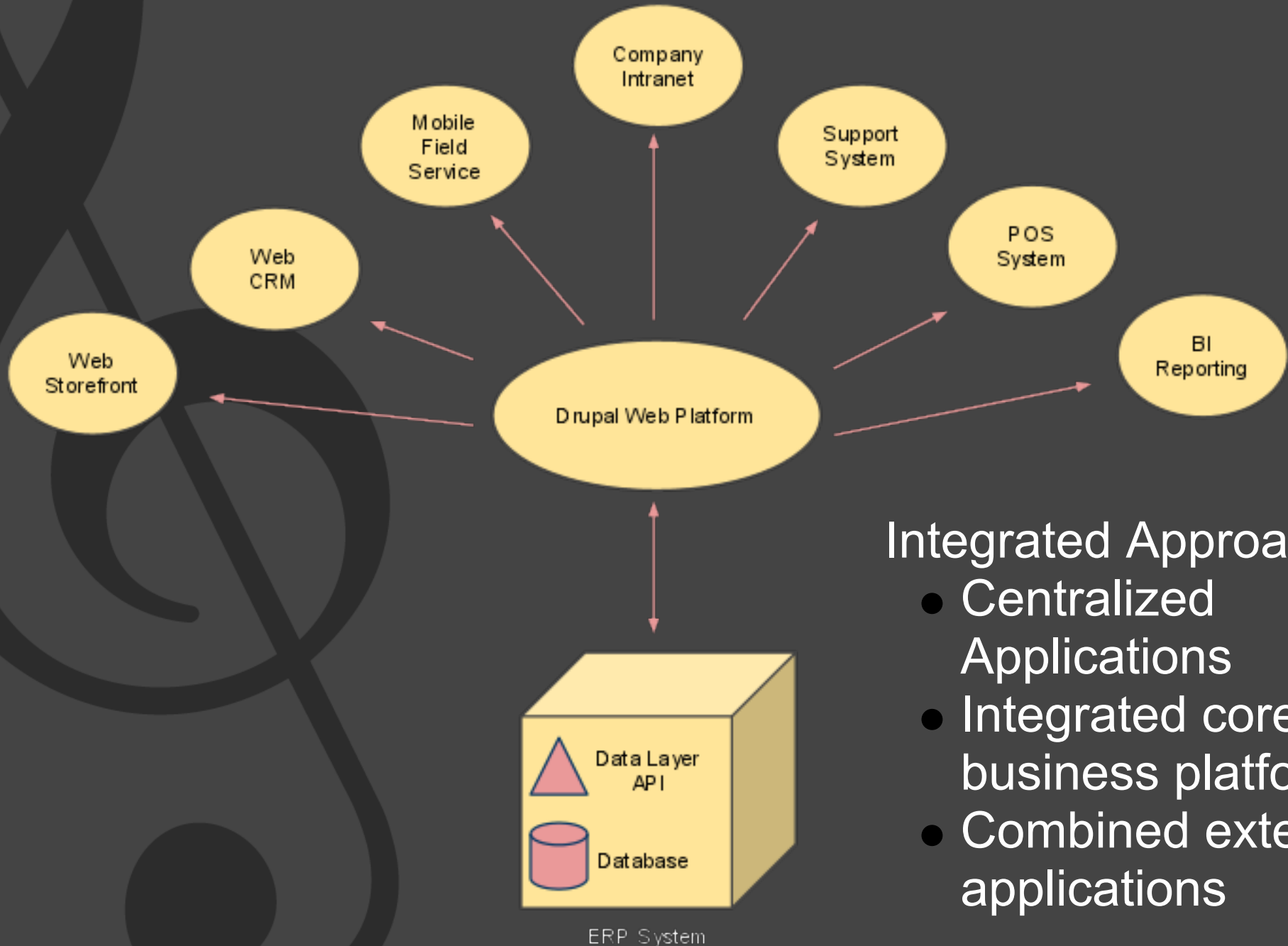
- Centralized Data
- Integrated core business functions
- Multiple external applications

How do Businesses Use CMS?



- Integrated Front-End Systems
 - Website
 - Intranet
 - Support System
 - E-commerce
 - CRM
 - Reporting
- Platform technology
 - Modular architecture
 - Extensible (usually through an API)
- Typically they have a **single database**
- Typically they **only** service customers or end-users

The Crazy Platform Integrated Model



Integrated Approach:

- Centralized Applications
- Integrated core business platforms
- Combined external applications

What is the Point?



- Back-end platforms (ERP systems) are used to fully integrate *back-end processes*
 - They are at the core of any business
- CMS Platforms (Drupal) help manage *diverse external applications*
 - These are the customer facing parts of the business
- Integration of *similar platforms* is more efficient than integration of applications
 - Integrations are traditionally application focused - this doesn't make sense!
 - Why integrate 10 applications when you could integrate 2 platforms?



What's the Problem?



PROBLEMS

NO MATTER HOW GREAT AND DESTRUCTIVE YOUR PROBLEMS MAY SEEM NOW,
REMEMBER, YOU'VE PROBABLY ONLY SEEN THE TIP OF THEM.

www.despair.com

Information richness

- Dilution via integration
- What **can** be integrated vs. What **should** be integrated

Requires a marketplace shift

- Move away from application integration
- Create cost justifications
- IT problems are often considered insurmountable

Platforms are hard to integrate

- Complexity = difficulty
- Competing information flows
 - what flows up vs. down



Smokey, This is Not 'Nam. This is Drupal. There are Rules.

Integration Rules to Live By (or at least bowl):



- "Gravity" = what goes up vs. what goes down
- "Inertia" = big objects/features are more difficult to move, but often provide the needed force to provide pathways for smaller ones
- "Relativity" = everything must be relative to one system - *unless you want the project to cost as much as the LHC*

How Do I Build This?



MACGYVER

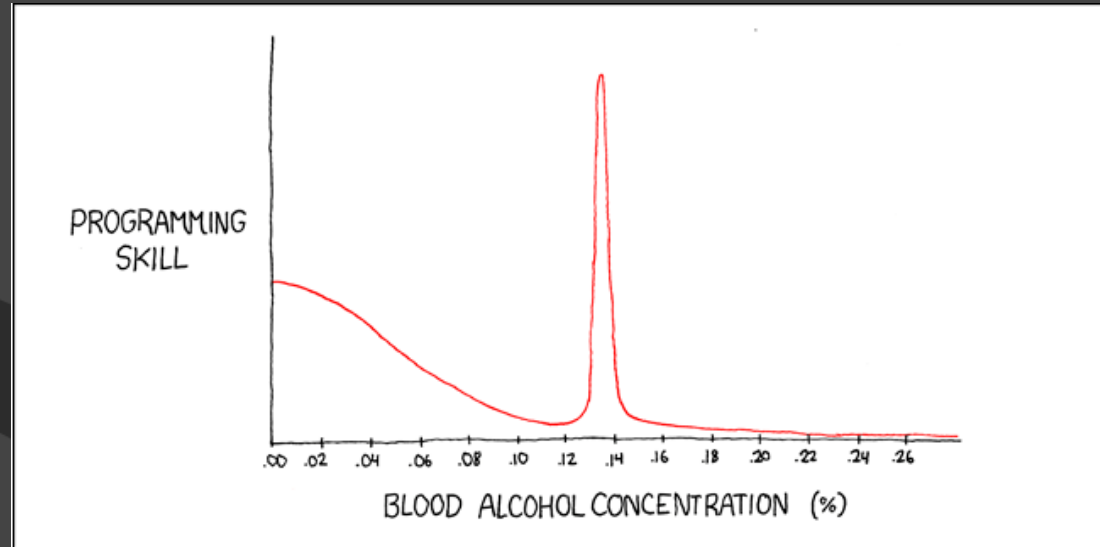
All he needed was a ball-point pen and a paper clip.

Key Components:

- Decide your integration points and directions
- Choose an integration method
- Develop some code
 - Contribute it
- Have a plan to maintain it
- Work on ways to extend it

Fundamentally, you're taking one database and periodically **connecting it** to another to share data over a **long period of time**. There are tons of ways to do this.

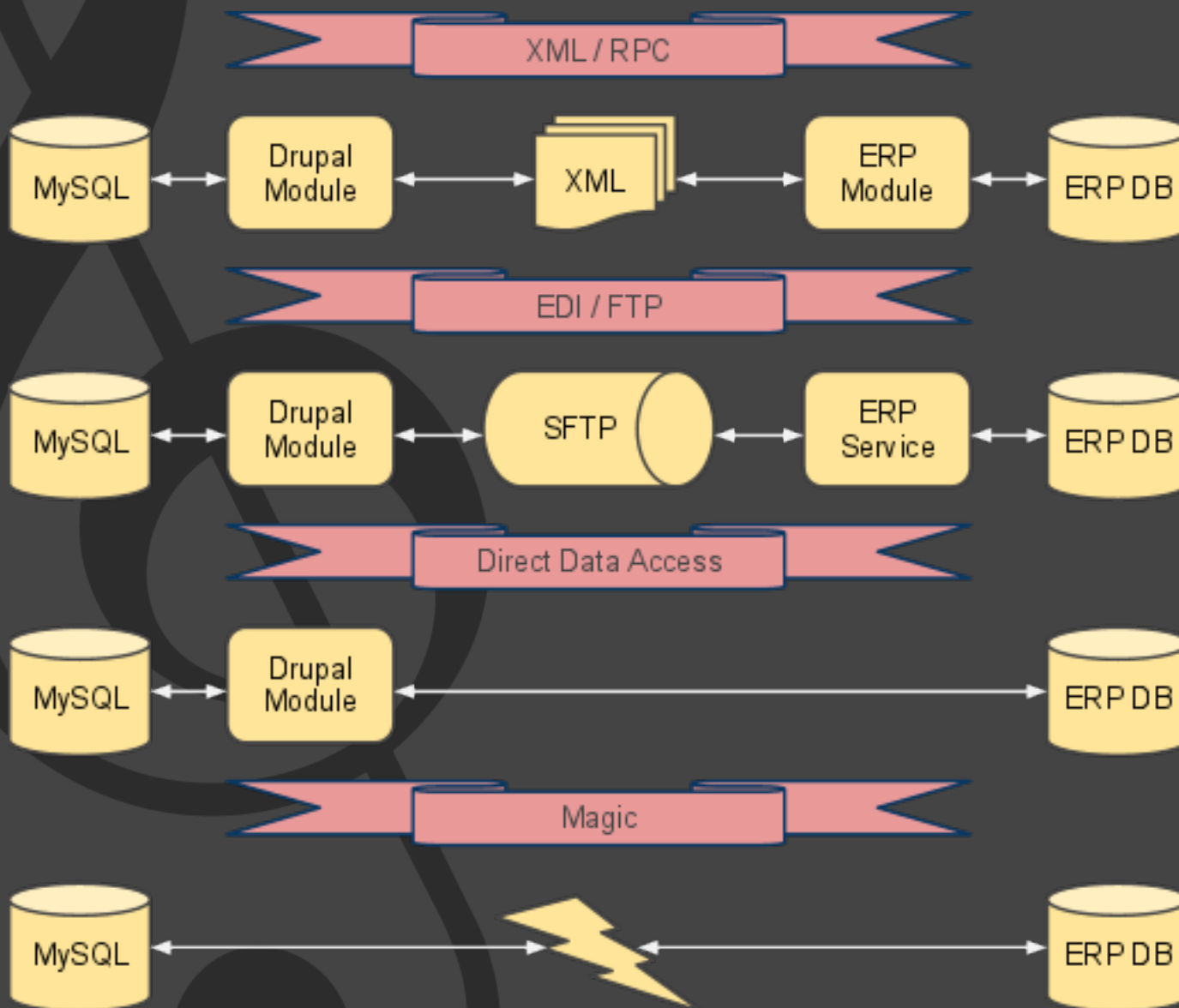
You'll Probably Actually Have to Write Some Code...



One of the hardest parts about performing an integration (other than knowing how much beer to buy) is figuring out exactly what Drupal needs to do to **fit within the framework of the ERP system.**

Typically you have to write a module or some custom code.

Your Ballpoint Pen and Paperclip



Best



Included Because I Wanted to Use a Lightning Bolt

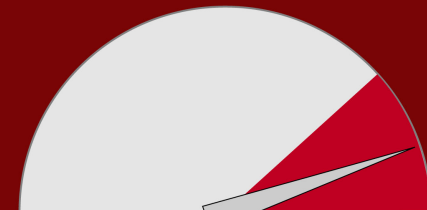
The Awesomness that is XML/RPC



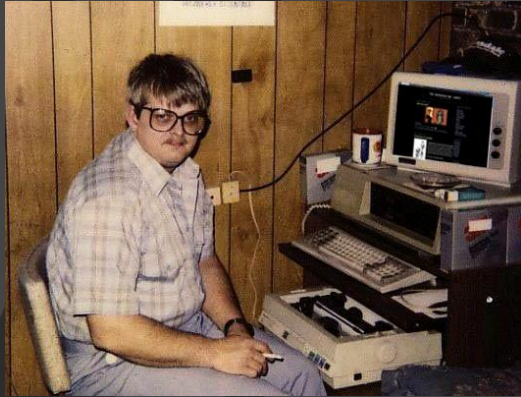
Awesome because:

- It is **easier** to perform bi-directional, real time integration
- Services / Feeds **framework exists** within Drupal
- Most **ERP systems support** XML export
- It is **easier to maintain** - all you have to do is update an XSLT
- Easy and **quick to extend** by just adding to the web service
- Easy to **perform logic on the data sent / received**

APPROACHING **CRITICAL** AWESOMENESS



EDI... is This the 90's?



Much like COBOL, it's hard to get rid of EDI, or even to talk about business systems without mentioning it.

- Spiffy **new EDI tools** make it not so old and busted
- Once it works, it almost always **continues to work** (unless something changes)
- Still allows for **relatively simple** bi-directional information flows in real-time
- Often will have some baseline **pre-existing standards**

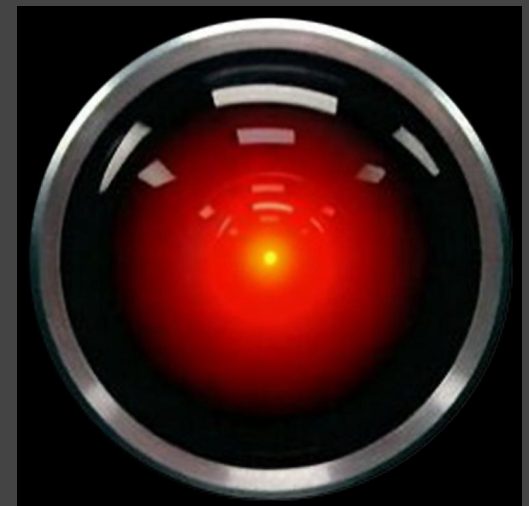
Direct Access = Direct Danger



In our opinion, direct database access should never be used for modification if an API exists!

- Still useful for **one-way integrations** (pulling data from a system)
- Typically **really easy** to connect to a database and get data out of it
- Often used for importing data that **doesn't frequently change** (i.e. monthly statements)

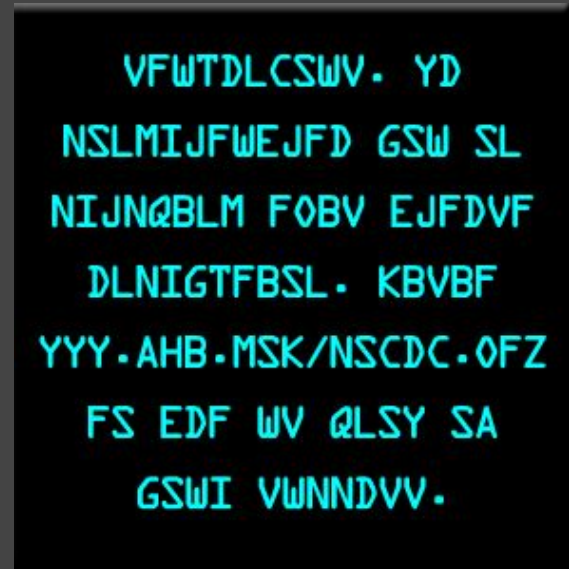
Hal wouldn't open the pod bay door because someone wrote the wrong update statement...



API's Are Cryptic for Most ERPs



Using an API for a system you don't really know is like trying to read Don Quixote in Pig Latin. You might be able to figure it out, but it's really silly and confusing.



It may be a good idea to call in an expert to help you out. Most firms have partners that implemented their ERP systems.

It is almost never acceptable to directly edit a database without using an API.

Middleware Can Be Your Messiah



Middleware applications exist for most of the major ERP systems. They're typically a lot **easier to learn** and use than the API of the system you're trying to integrate.

Many also help **prevent many of the coding sins** that you could commit when programming for a specific application.

Middleware can work miracles...





Consider the Other Side Before You Jump In

- ERP Systems are **complicated!**
- Be sure that you **communicate** with the company that implemented the ERP
- Code may need to be written for the ERP to **provide users or administrators with management tools**



Key Points



1. Ubiquity of ERP in business
2. Influence of ERP on business processes
3. Progression to platform integration from application integration
4. Drupal/CMS as front end management
5. No single "right way" to perform an integration
6. Tailor your approach to a technology



ORCHESTRA
TEAM

Q & A

Contact us:

www.orchestrateam.com

Brad Windecker

brad@orchestrateam.com

www.twitter.com/OrchestraTeam

Chris Strahl

strahl@orchestrateam.com

www.twitter.com/chrisstrahl