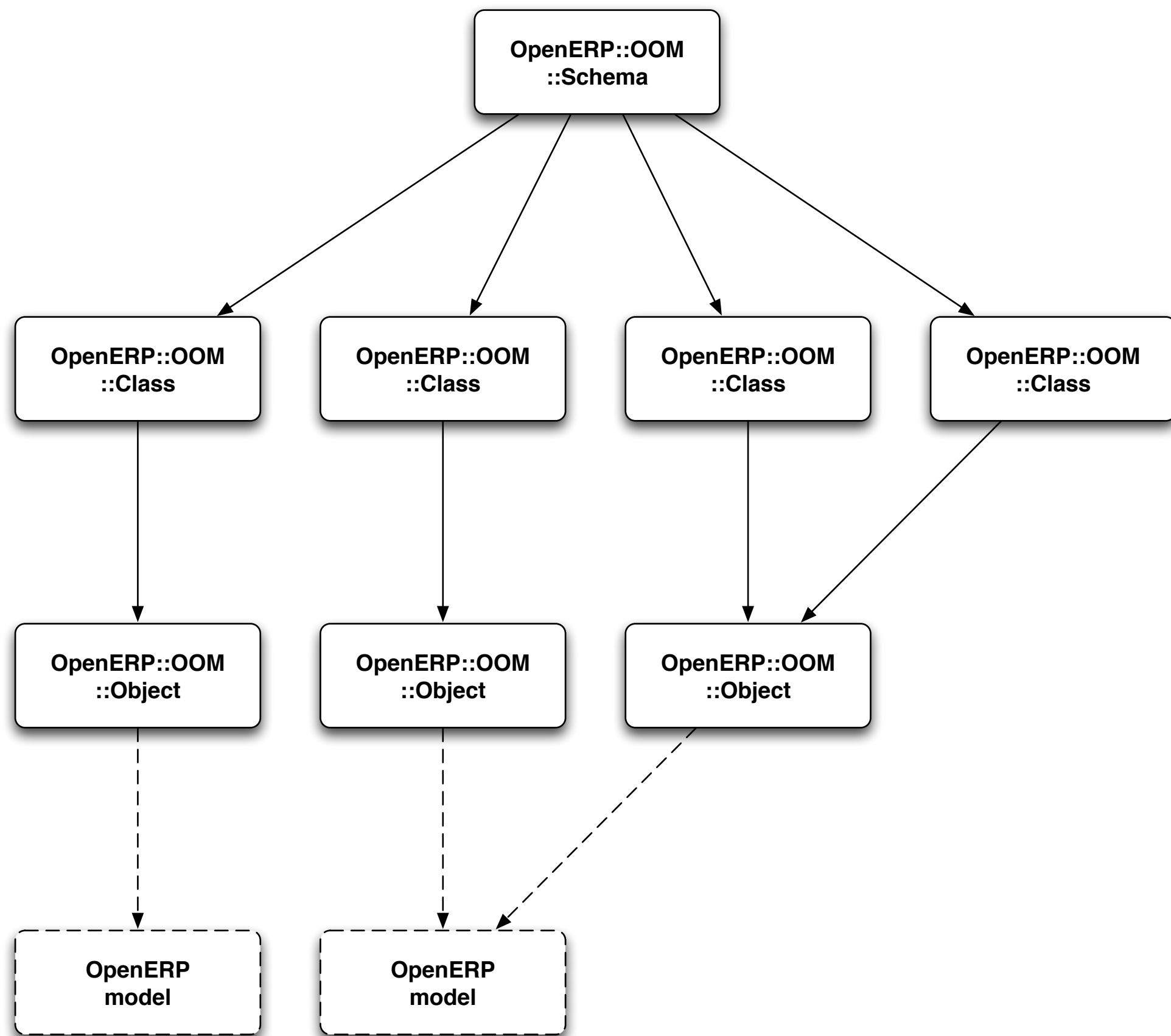
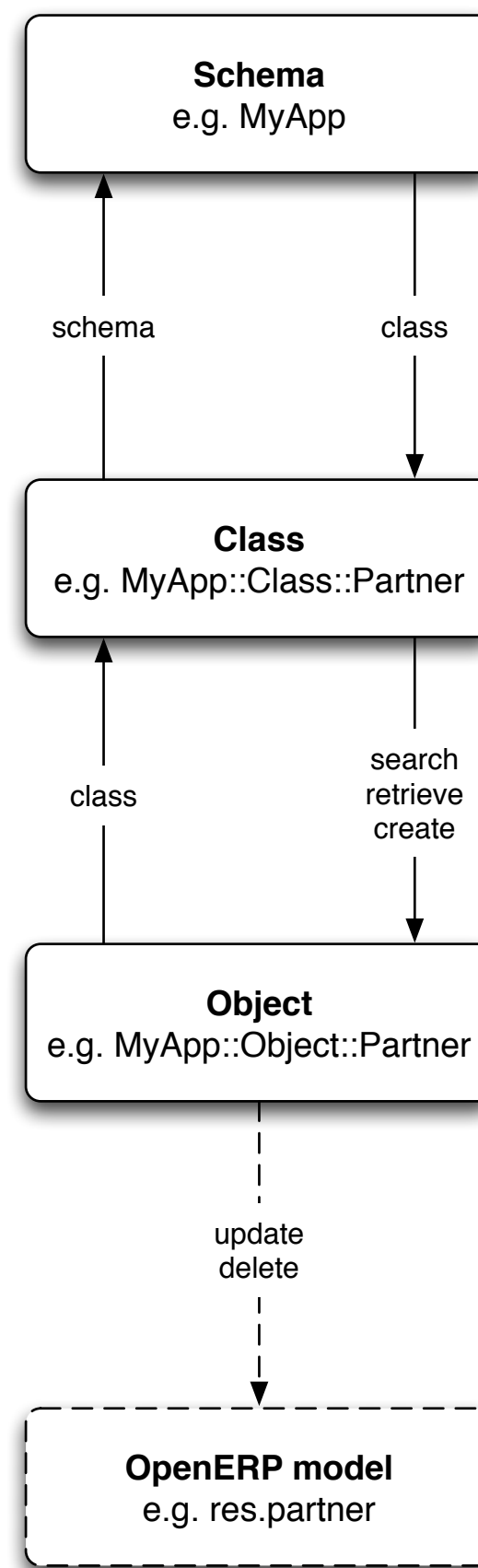


Object model



Programming model



```
# example.pl

use 5.010;
use strict;
use warnings;
use MyApp;

my $schema = MyApp->new(
    openerp_connect => {
        host      => '172.16.128.129',
        dbname    => 'jj_test_1',
        username  => 'admin',
        password  => 'admin',
    },
);

# Create object
if (my $company = $schema->class('Company')->create({name=>"My Company"})) {
    say "Created company with ID " . $company->id;

    # Update object
    $company->name("My New Name");
    $company->update;

    # Delete object
    $company->delete;
}

# Search objects
foreach my $company ($schema->class('Company')->search) {
    say $company->id . ": " . $company->name;

    # Follow relationship
    say $company->partner->name;
};
```

File layout

Name	Date Modified	Size	Kind
MyApp.pm	Today, 10:50	4 KB	Perl File
MyApp	Today, 17:18	--	Folder
Object	Today, 16:14	--	Folder
Partner.pm	Today, 15:43	4 KB	Perl File
Company.pm	Today, 16:14	4 KB	Perl File
Class	Today, 16:14	--	Folder
Partner.pm	Today, 15:43	4 KB	Perl File
Company.pm	Today, 16:14	4 KB	Perl File

8 items, 42.81 GB available

```
# MyApp.pm

package MyApp;

use Moose;
extends 'OpenERP::OOM::Schema';

1;
```

```
# MyApp/Class/Company.pm

package MyApp::Class::Company;
use OpenERP::OOM::Class;

object_type 'MyApp::Object::Company';

1;
```

Add class methods here,
e.g. for canned searches

```
# MyApp/Object/Company.pm

package MyApp::Object::Company;
use OpenERP::OOM::Object;

openerp_model 'res.company';

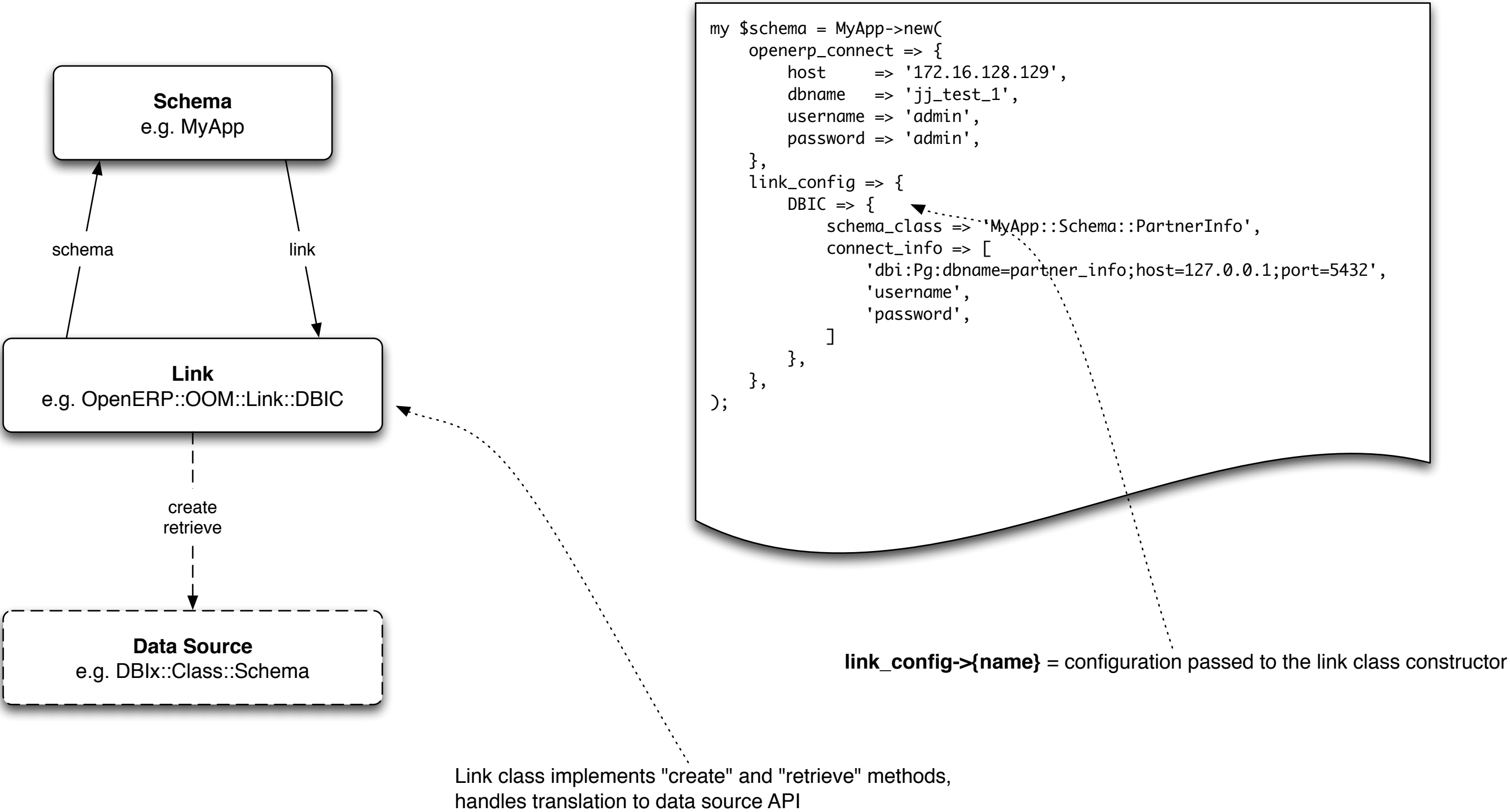
has 'name'      => (isa=>'Str', is=>'rw');
has 'account_no' => (isa=>'Str', is=>'rw');

relationship 'partner' => (
    key   => 'partner_id',
    type  => 'manyZone',
    class => 'Partner',
);

1;
```

Add object methods here, i.e.
business logic relating to that object

Relationship = join between two OpenERP::OOM::Object types
Link = join between an OpenERP::OOM::Object and an OpenERP::OOM::Link class (i.e. external data source)



```
# MyApp/Object/Company.pm

package MyApp::Object::Partner;
use OpenERP::OOM::Object;

openerp_model 'res.partner';

has 'name' => (isa=>'Str', is=>'rw');

relationship 'addresses' => (
    key    => 'address',
    type   => 'one2many',
    class  => 'PartnerAddress',
);

has_link 'extra_info' => (
    key    => 'x_notes_id',
    type   => 'single',
    class  => 'DBIC',
    args   => { class => 'Notes' }
);

1;
```

```
# example.pl

my $partner = $schema->class('Partner')->retrieve($id);

# Create a linked object
$partner->create_related('extra_info', {
    notes => $note,
});

say $partner->id . ": " . $partner->name;

# Follow the link
say $partner->extra_info->notes;

1;
```

has_link 'name' = name for the object method to follow the link
key = OpenERP model field to store foreign key(s)
type = "single" (one to one relationship) or "multiple" (one to many relationship)
class = OpenERP::OOM::Link class
args = Arguments to pass to the link class "create" and "retrieve" methods