

```
//destination side
class Link :
{
private:
    event_handler* eh_;
public:
    void deliver(event* ev)
    {
        eh_->handle(ev);
    }
};
```

```
//source side
class Link :
{
public:
    void send(event* ev)
    {
        ...
    }
};
```

Discrete  
Event  
Core  
Magic

```
template <class Cls, class Function>
class member_fxn_handler :
    public event_handler
{
private:
    Cls* base_; //Listener in example
    Function* fxn_; //Listener::recv in example
public:
    void handle(event* ev){
        (base_->*fxn_)(ev); //dispatch
    }
};
```

```
class Listener :
public event_component {
public:
    ...
    void recv(event* ev){
        ....
    }
};
```

```
class Pinger :
public event_component {
private:
    Link* link_;
public:
    void send(){
        auto ev = new event(...);
        link_->send(ev);
    }
};
```

