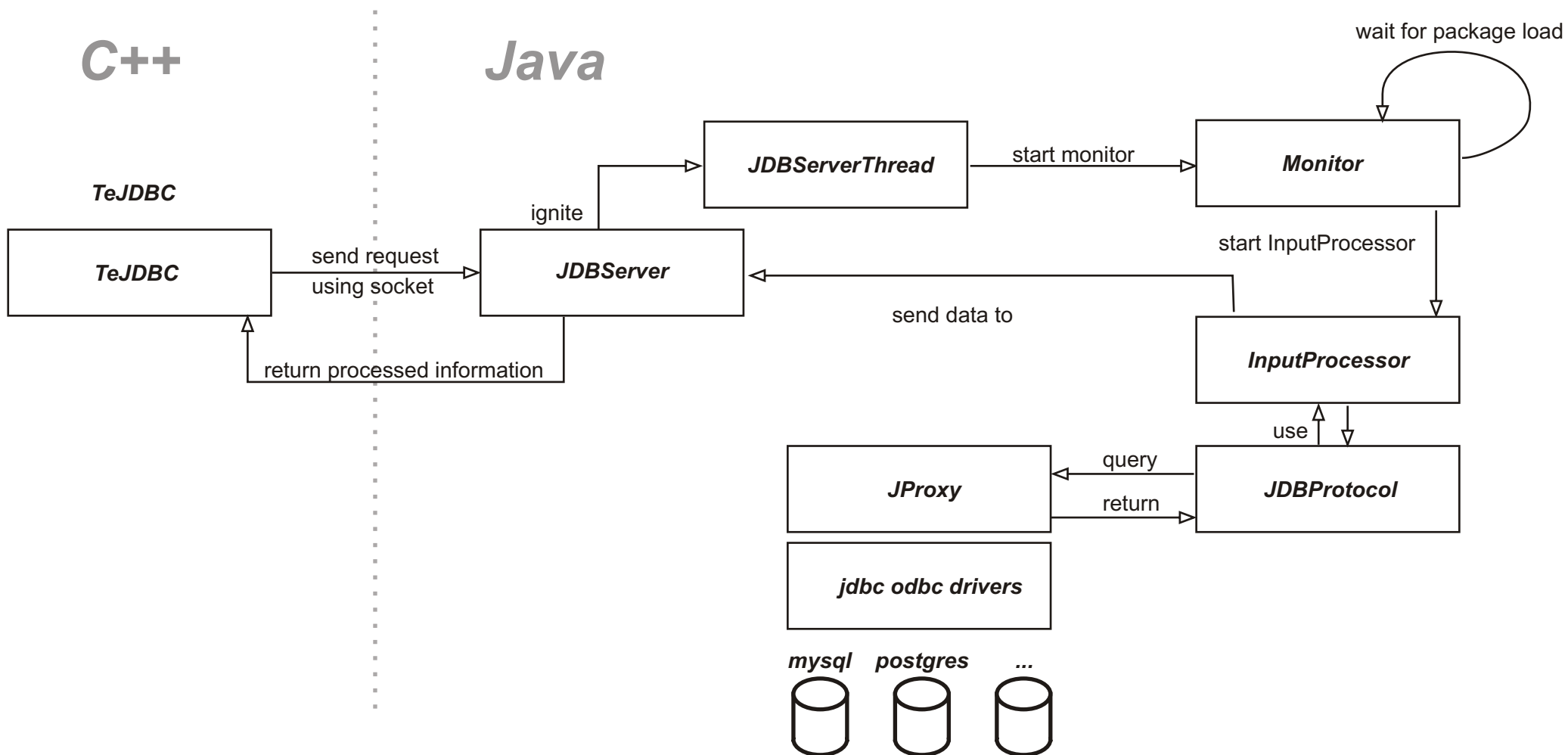




*Driver Genérico para Acesso a Banco de Dados*  
*Eduardo Sant'Ana da Silva*

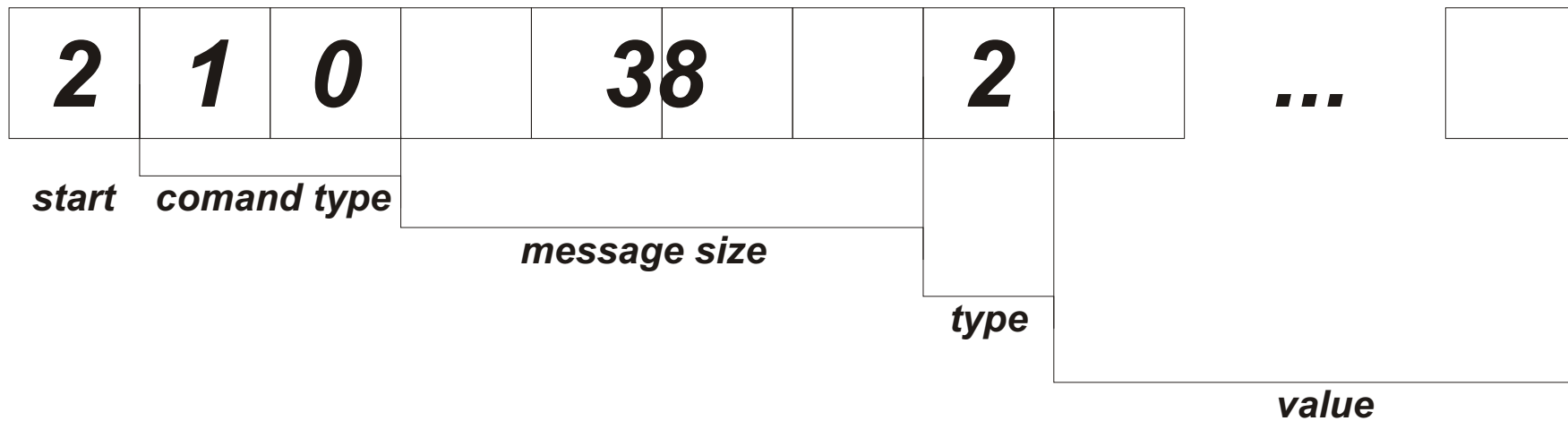


# Modelo Conceitual



# Protocolo (versão alpha)

ex:



# Codificação de Tipos

```
public static byte[] encodeLong(long i) {  
    byte[] b = new byte[8];  
    b[0] = (byte) ((i) & 0xFF);  
    i >>= 8;  
    b[1] = (byte) ((i) & 0xFF);  
    i >>= 8;  
    b[2] = (byte) ((i) & 0xFF);  
    i >>= 8;  
    b[3] = (byte) ((i) & 0xFF);  
    i >>= 8;  
    b[4] = (byte) ((i) & 0xFF);  
    i >>= 8;  
    b[5] = (byte) ((i) & 0xFF);  
    i >>= 8;  
    b[6] = (byte) ((i) & 0xFF);  
    i >>= 8;  
    b[7] = (byte) ((i) & 0xFF);  
    i >>= 8;  
    return b;  
}
```

# Codificação de Tipos

```
public static byte[] encodeInt(int i) {  
    byte[] b = new byte[4];  
    b[0] = (byte) ((i) & 0xFF);  
    i >>= 8;  
    b[1] = (byte) ((i) & 0xFF);  
    i >>= 8;  
    b[2] = (byte) ((i) & 0xFF);  
    i >>= 8;  
    b[3] = (byte) ((i) & 0xFF);  
    i >>= 8;  
    return b;  
}
```

# Codificação de Tipos

```
public static byte[] encodeShort(int i) {  
    byte[] b = new byte[2];  
    b[0] = (byte) ((i) & 0xFF);  
    i >>= 8;  
    b[1] = (byte) ((i) & 0xFF);  
    i >>= 8;  
    return b;  
}
```

# Decodificação de Tipos

```
public static int getInt(byte[] b) {  
    int l = 0;  
    l |= b[3] & 0xFF;  
    l <<= 8;  
    l |= b[2] & 0xFF;  
    l <<= 8;  
    l |= b[1] & 0xFF;  
    l <<= 8;  
    l |= b[0] & 0xFF;  
    return l;  
}
```

# Decodificação de Tipos

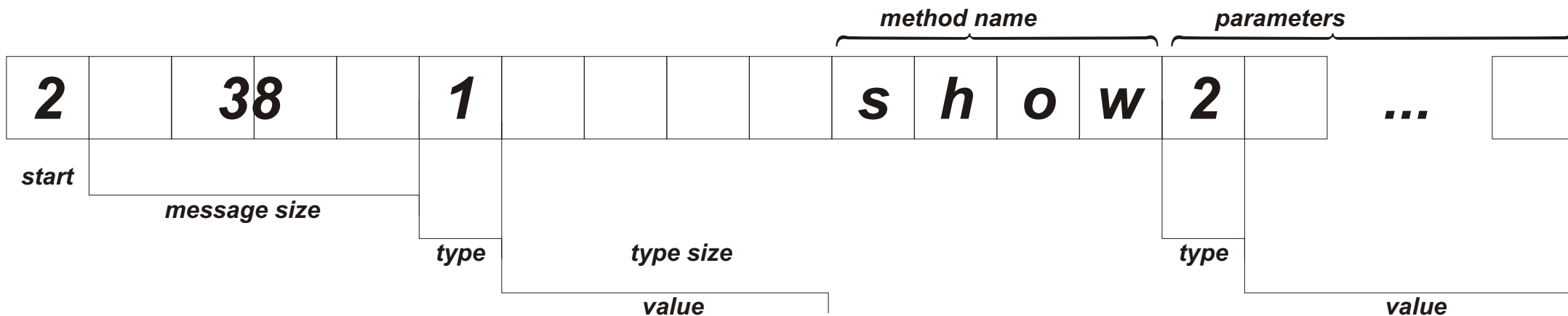
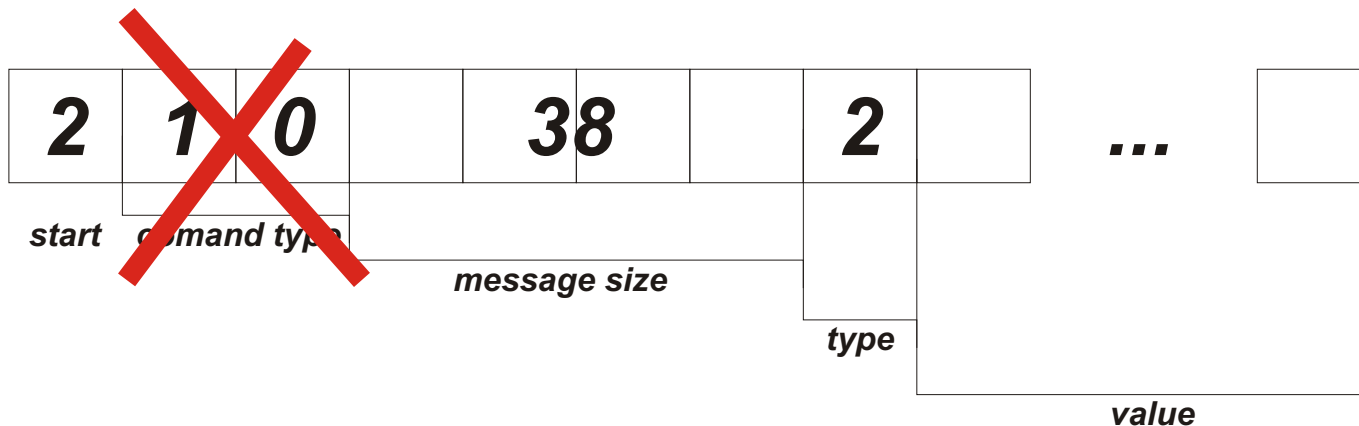
```
public static short getShort(byte[] b) {  
    short s = 0;  
    s |= b[1] & 0xFF;  
    s <<= 8;  
    s |= b[0] & 0xFF;  
    return s;  
}
```



# Decodificação de Tipos

```
public static long getLong(byte[] b) {  
    long l = 0;  
    l |= b[7] & 0xFF;  
    l <<= 8;  
    l |= b[6] & 0xFF;  
    l <<= 8;  
    l |= b[5] & 0xFF;  
    l <<= 8;  
    l |= b[4] & 0xFF;  
    l <<= 8;  
    l |= b[3] & 0xFF;  
    l <<= 8;  
    l |= b[2] & 0xFF;  
    l <<= 8;  
    l |= b[1] & 0xFF;  
    l <<= 8;  
    l |= b[0] & 0xFF;  
    return l;  
}
```

# Protocolo (nova versão)



# Benefícios da Nova Versão do Protocolo

**Não há mais necessidade do Mapeamento do "Command Type" para um método ex:**

```
switch (commandMessage) {  
    case msgShowDatabasesResponse:  
        processaShowDatabasesResponse(parameters);  
        break;  
}
```

**Utilizando-se a nova versão o método a ser chamado será um dos parâmetros informados, não havendo necessidade assim de alteração do protocolo caso algum método novo seja implementado.**

**O único código a ser mudado será o JProxy que acessa o banco de dados.**

# Tarefas a serem executadas

**. aRT**

**a) Fazer uma versão sem qualquer referência a conexão do banco de dados, sem guardar cache nem algo do tipo.**

**b) Fazer um script de teste que utilize todos os métodos presentes no TeMySQL ( utilizado como base para o primeiro TeJDBC driver ).  
O script deve utilizar um banco de dados com valores pré-definidos e testar os resultados com um teste unitário para cada método.**

# Fim



outros drivers