

VFS and Block I/O

VA Linux Systems Japan

Hiroataka.Sasaki

sasaki@valinux.co.jp

目次

- ファイルシステム概要
- VFS (仮想ファイルシステム)
- ページキャッシュとバッファキャッシュ
- ブロックI/O共通レイヤ

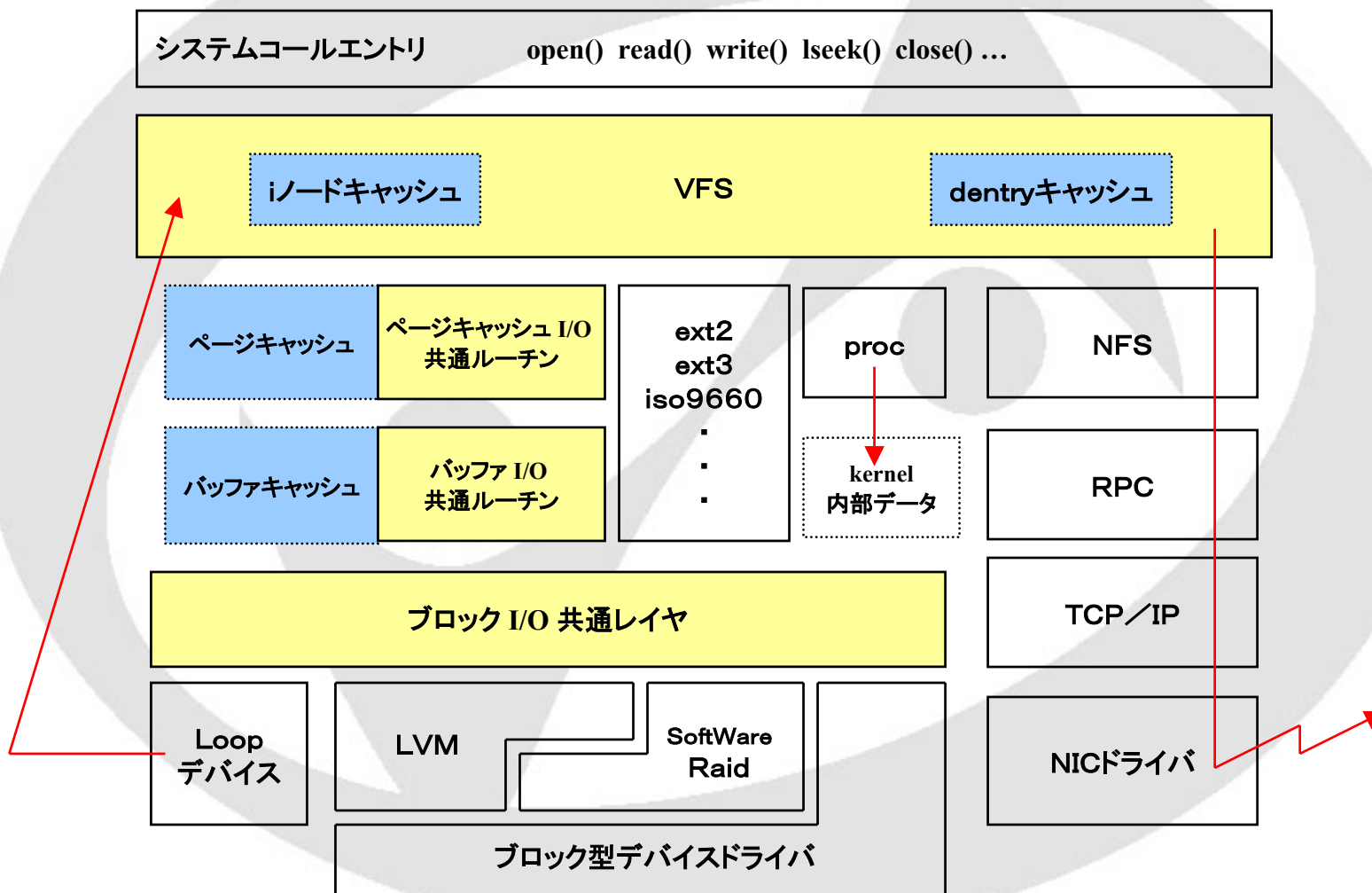
ファイルシステム概要

ファイルシステムとは？

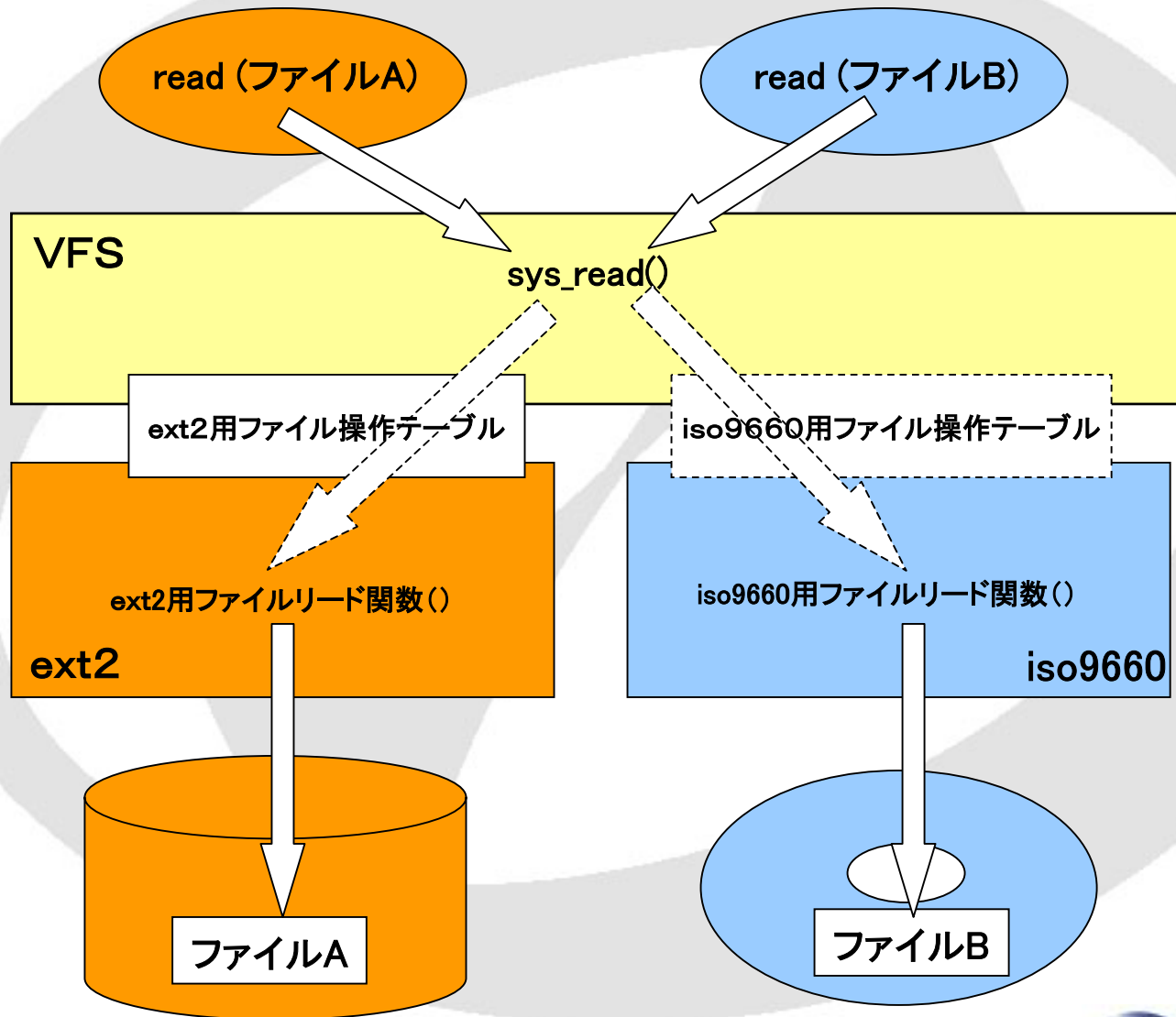
記憶デバイス上のデータをファイルという論理的な構造で抽象化し利用するための機構

- 論理的なファイルと物理的なディスクブロックの対応を管理
- ファイルに対するアクセス要求を、ディスクブロックへのアクセス要求へ変換
- ファイルの生成、拡張、削除に応じた、ディスクブロックの割り当て

ファイルシステムのレイヤ構造



VFSの概要



さまざまなファイルシステム

- ローカルファイルシステム
 - ー ローカルディスク上のファイルシステム
 - ー Linuxの標準はext2ファイルシステム
 - ー 他、FFS(BSD)、VFAT(Win98)、iso9660など多数
 - ー ext3、XFSなどのジャーナリングFSもサポート
- ネットワークファイルシステム
 - ー ネットワークの先にあるファイルシステムを共有
 - ー 代表的(Unix系)なものはNFS
 - ー 他、CODA、SMB、NCP、Intermezzoなど
- 特殊なファイルシステム
 - ー procファイルシステム
 - ー kernel内部データをファイルとして抽象化

ブロックI/O共通レイヤ

- ディスクブロックのアクセス要求をもとに
ブロック型デバイスドライバを起動
- ファイルシステム構造には関与しない
- バッファI/O要求のクラスタリング

さまざまなキャッシュ

キャッシュによる処理の高速化

- ページキャッシュ
(ファイルデータのキャッシュ)
- バッファキャッシュ
(ディスクブロックのキャッシュ)
- iノードキャッシュ
- dentryキャッシュ



VFSのデータオブジェクト

- `file_system_type`オブジェクト
- `super_block`オブジェクト
- `inode`オブジェクト
- `dentry`オブジェクト
- `file`オブジェクト
- `address_space`オブジェクト

iノードオブジェクト

```
struct inode {  
    iノード番号;  
    利用カウンタ;  
    デバイス番号;  
    ハードリンク数;  
    :
```

```
    union {  
        minix固有のiノード情報;  
        ext2固有のiノード情報;  
        :  
    }  
}
```

minix固有のiノード情報

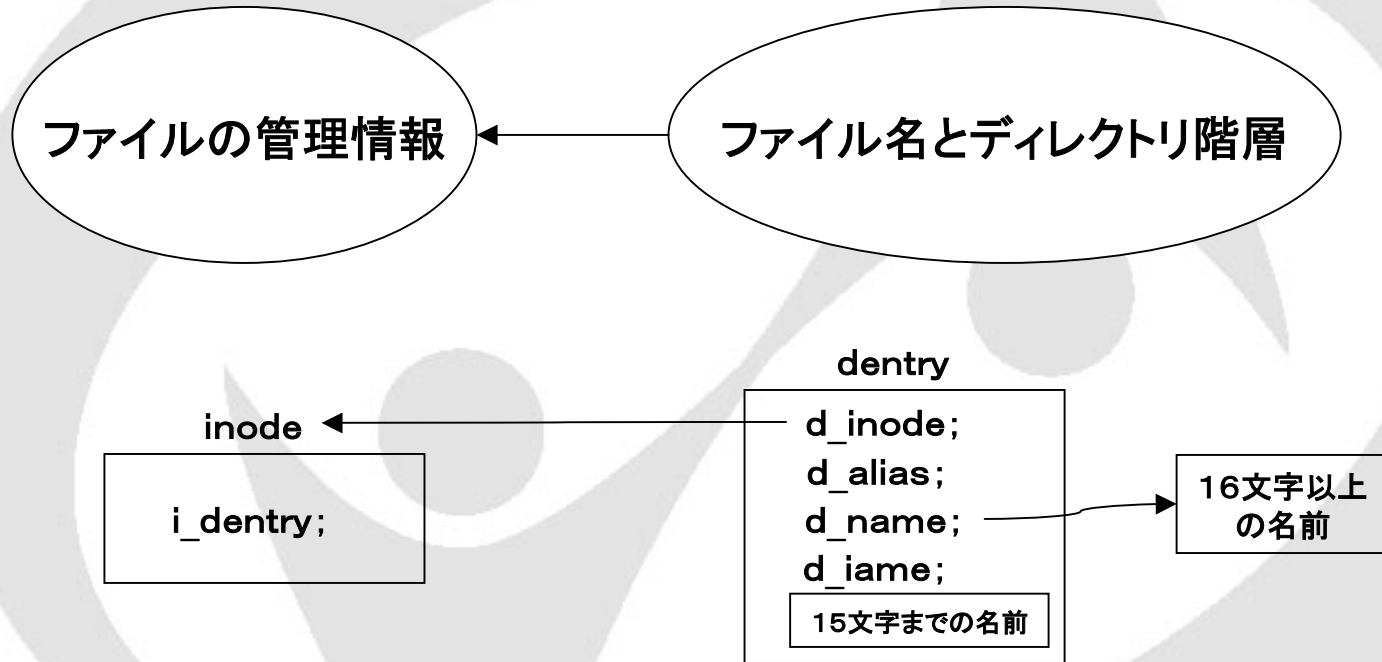
ext2固有のiノード情報

ext3固有のiノード情報

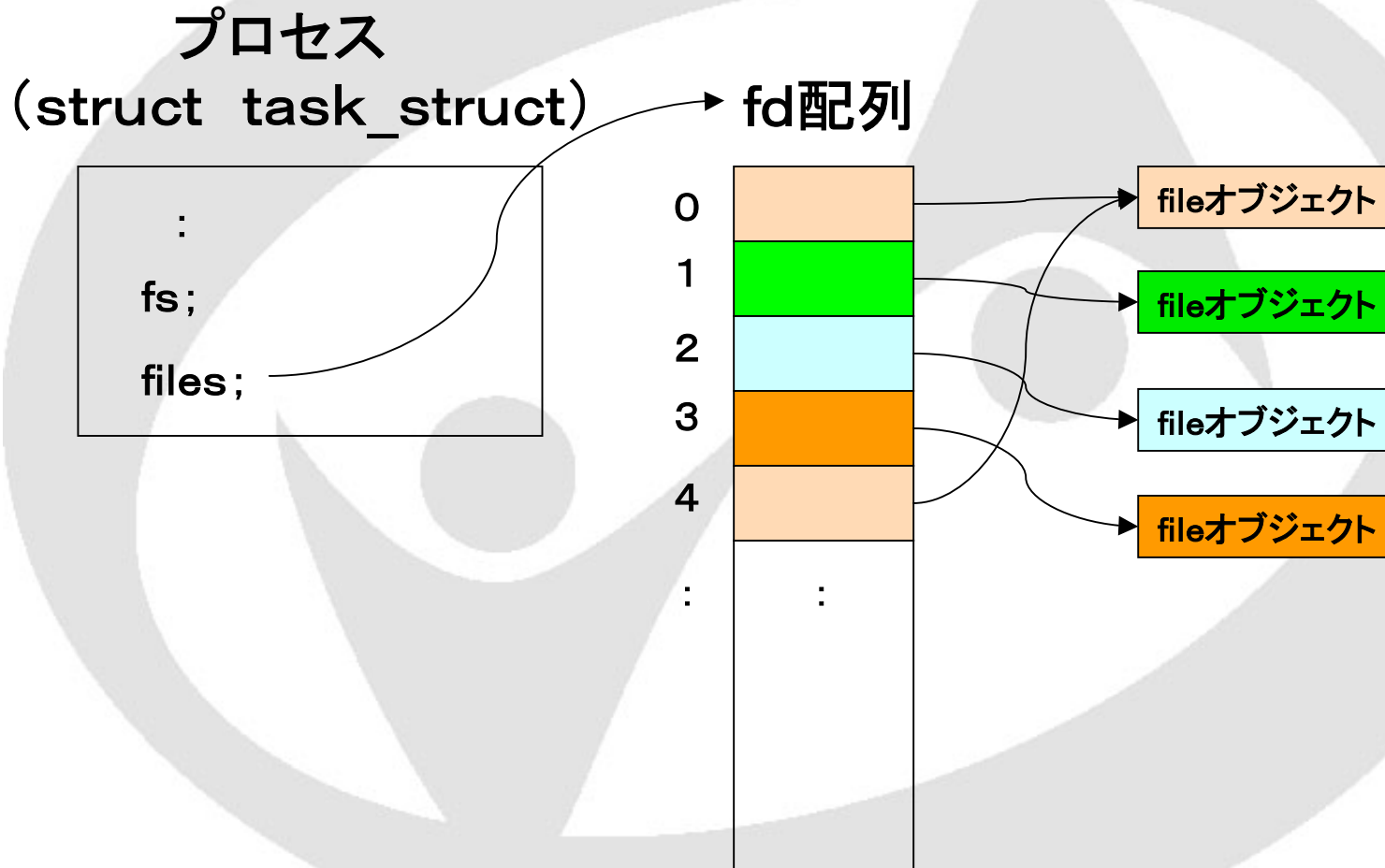
nfs固有のiノード情報

:

iノードオブジェクトとdentryオブジェクト



プロセスとファイル

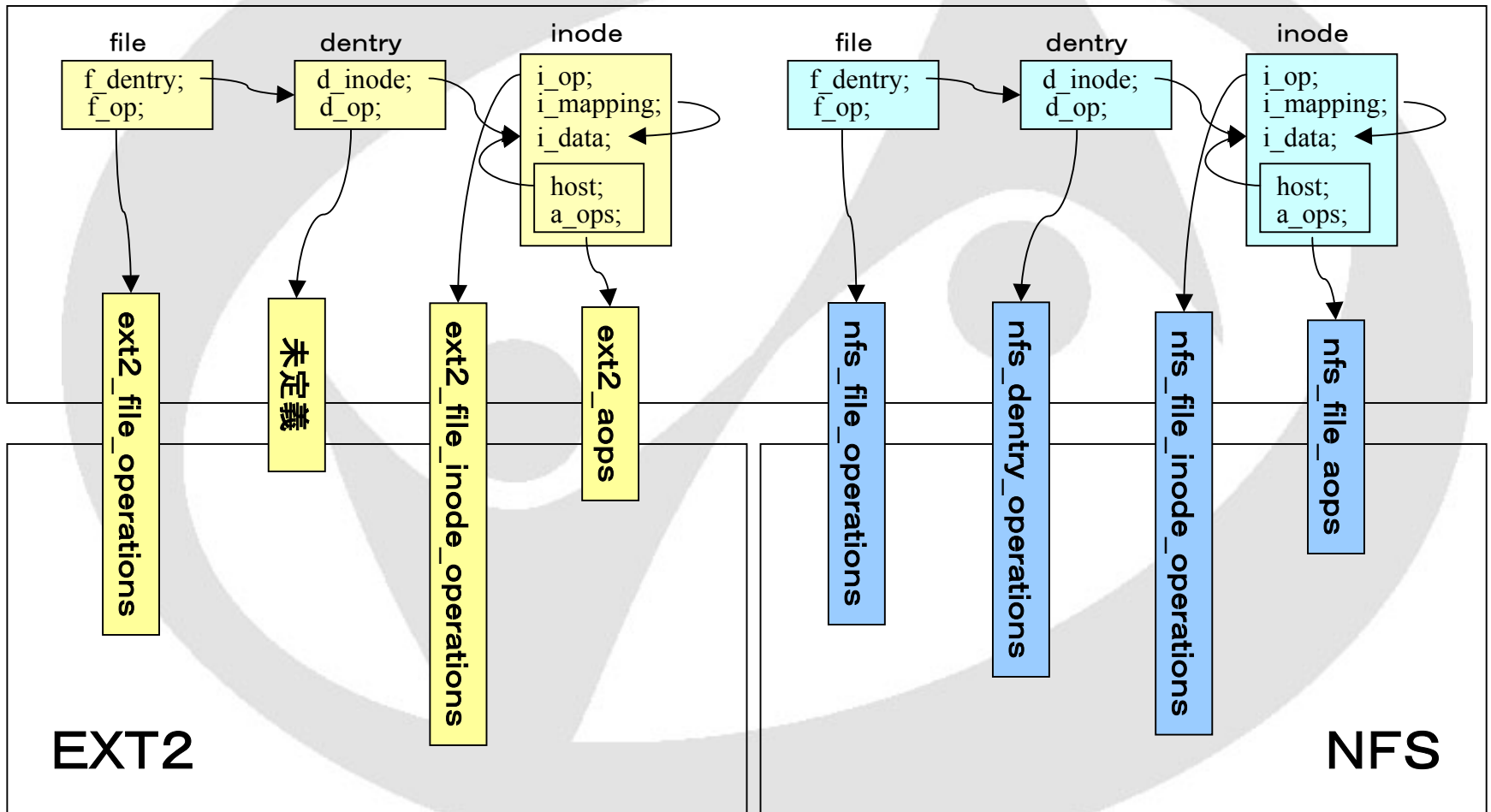


VFSとファイルシステムのI/F

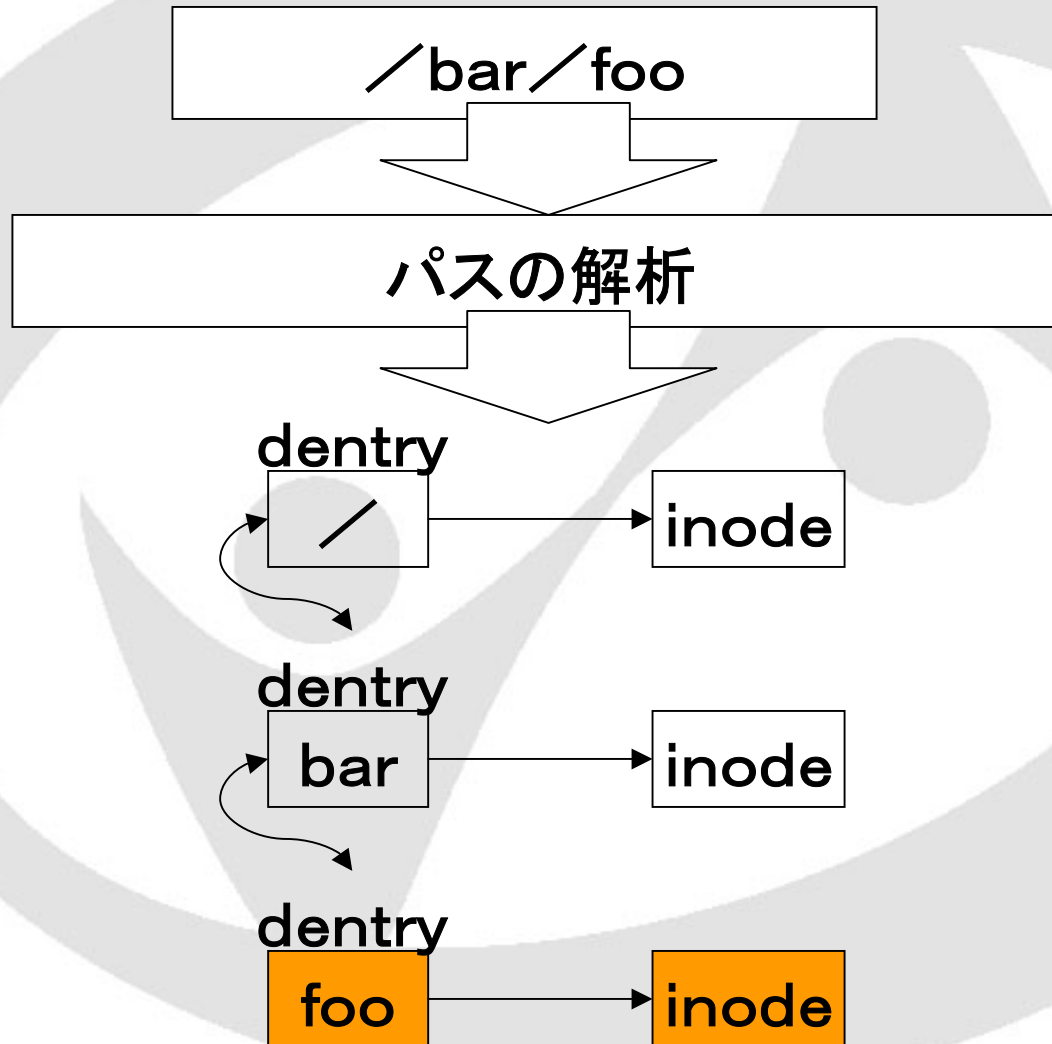
- `super_block`読み込みメソッド
(`file_system_type`オブジェクトの`read_super`メンバ)
- `super_operations`テーブル
(`super_block`オブジェクトの`s_op`メンバ)
- `inode_operations`テーブル
(`inode`オブジェクトの`i_op`メンバ)
- `dentry_operations`テーブル
(`dentry`オブジェクトの`d_op`メンバ)
- `file_operations`テーブル
(`file`オブジェクトの`f_op`メンバ)
- `address_space_operations`テーブル
(`address_space`オブジェクトの`a_ops`メンバ)

VFSとファイルシステムのI/F(通常ファイル)

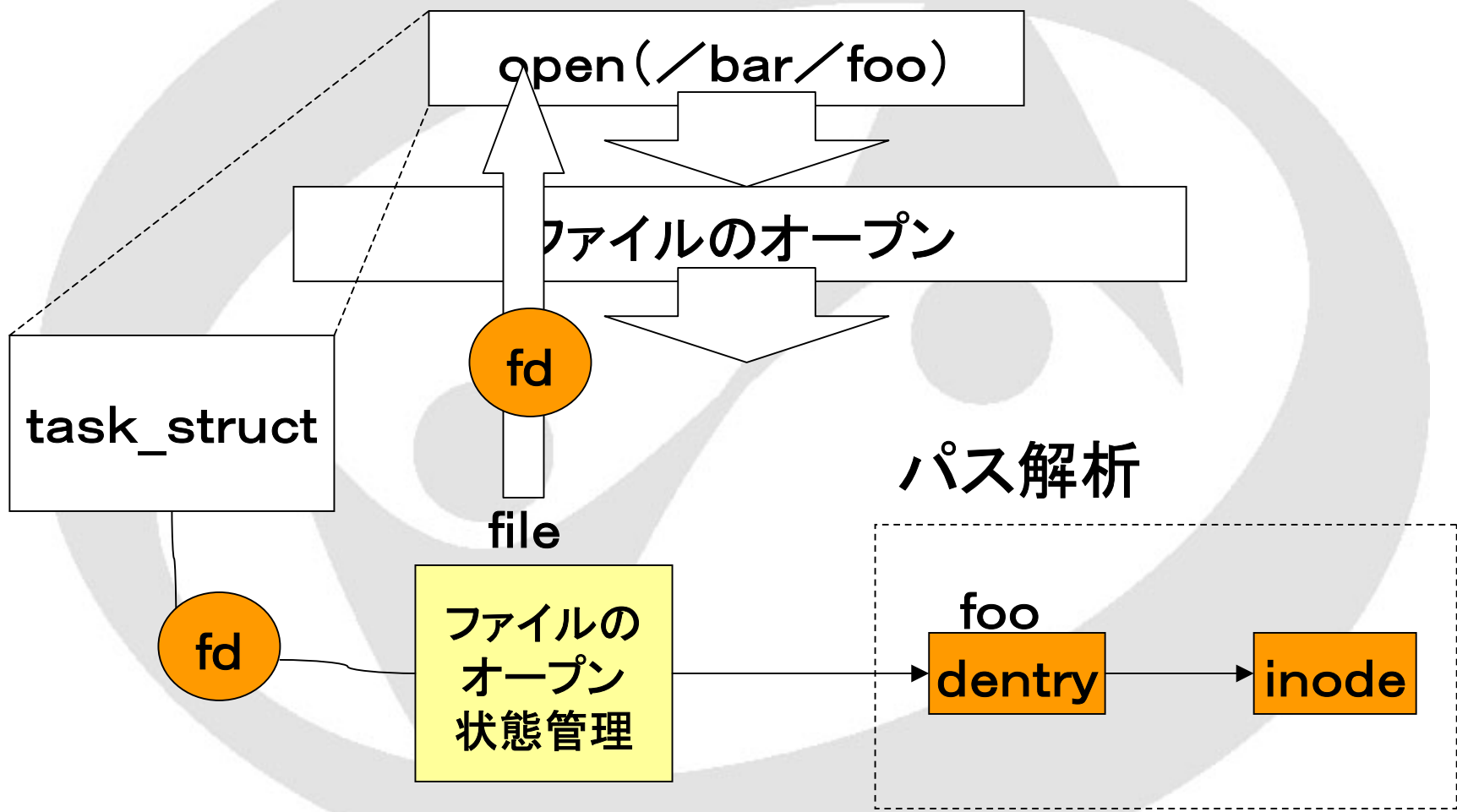
VFS



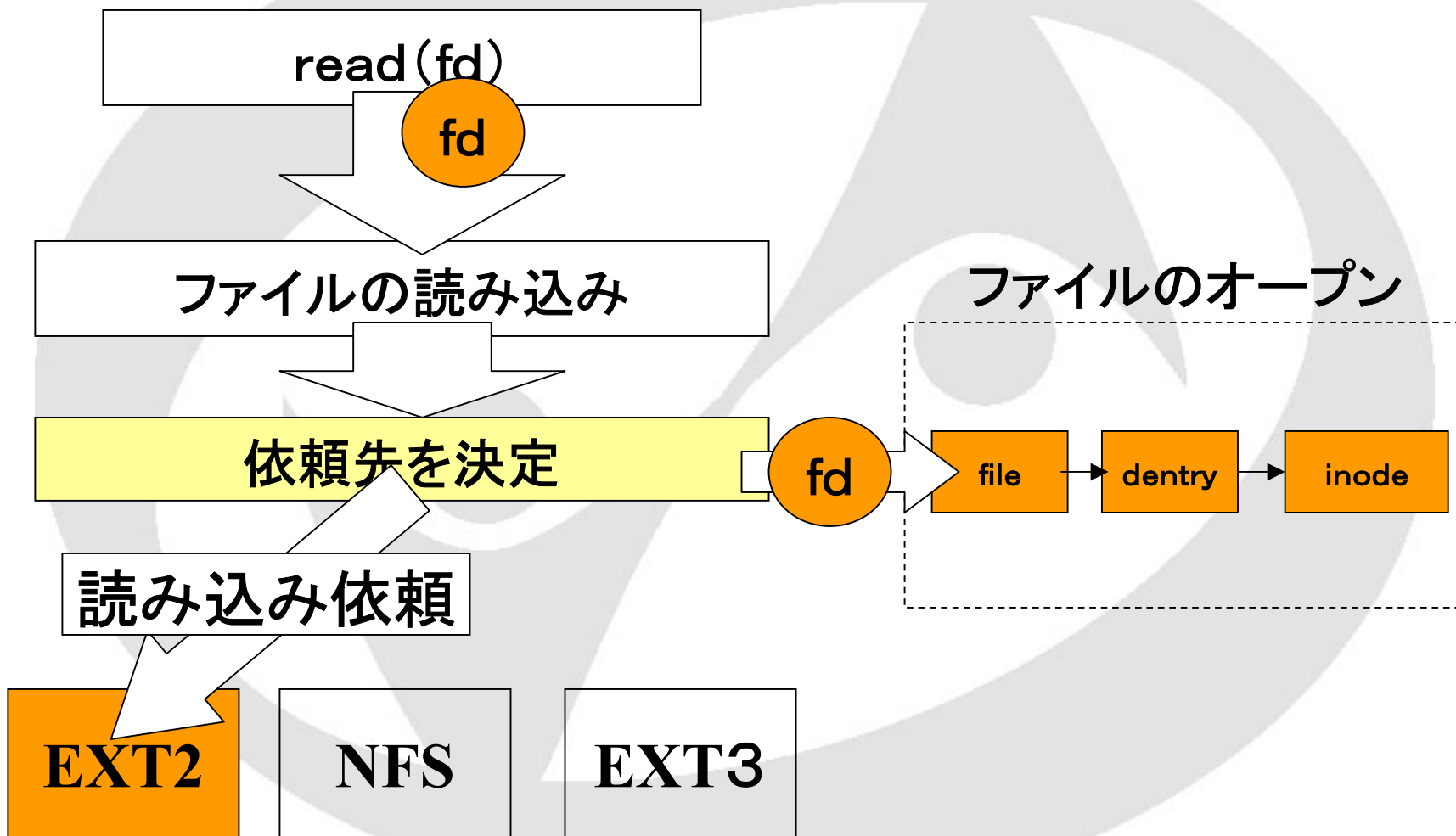
パスの解析



ファイルのオープン



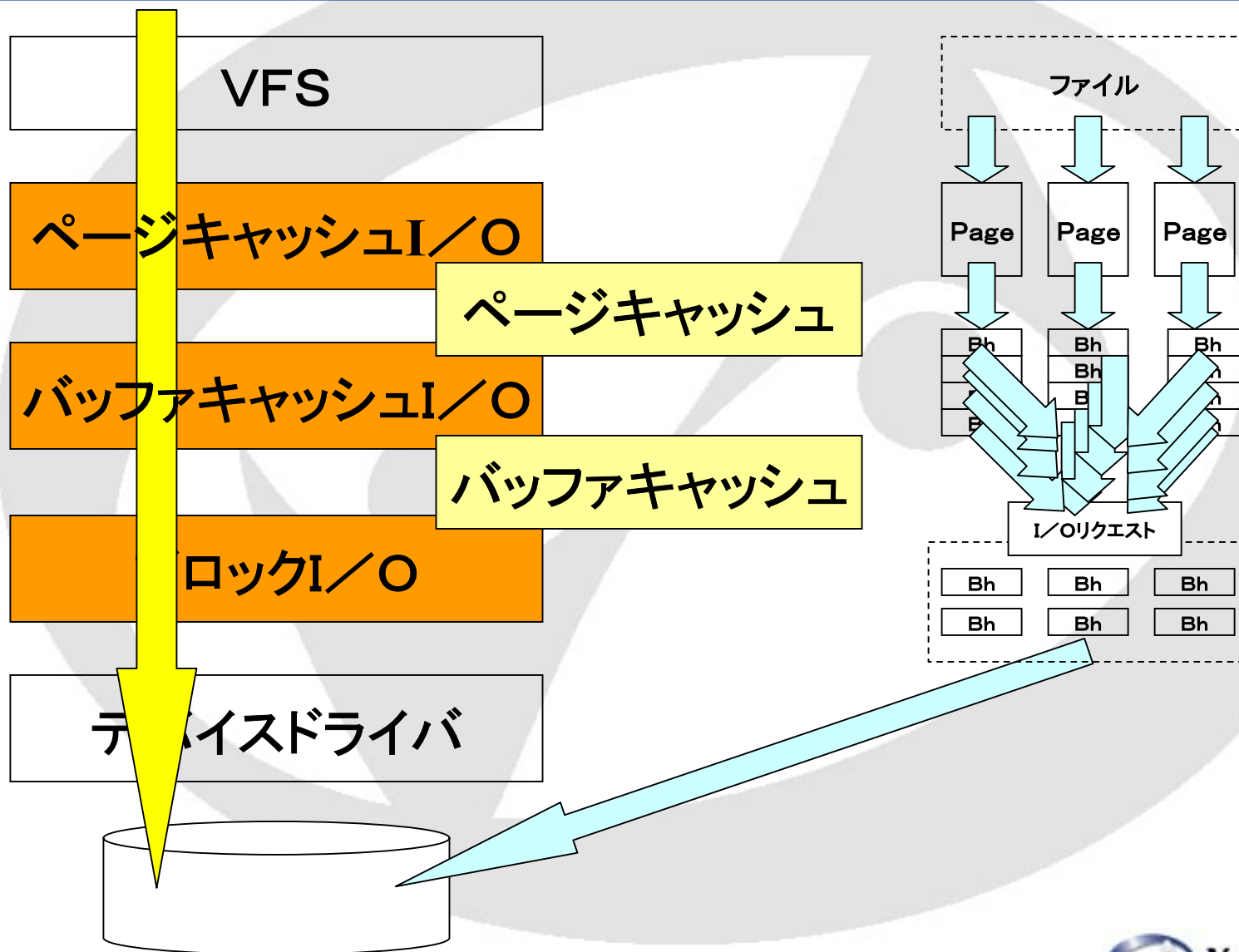
ファイルの読み込み



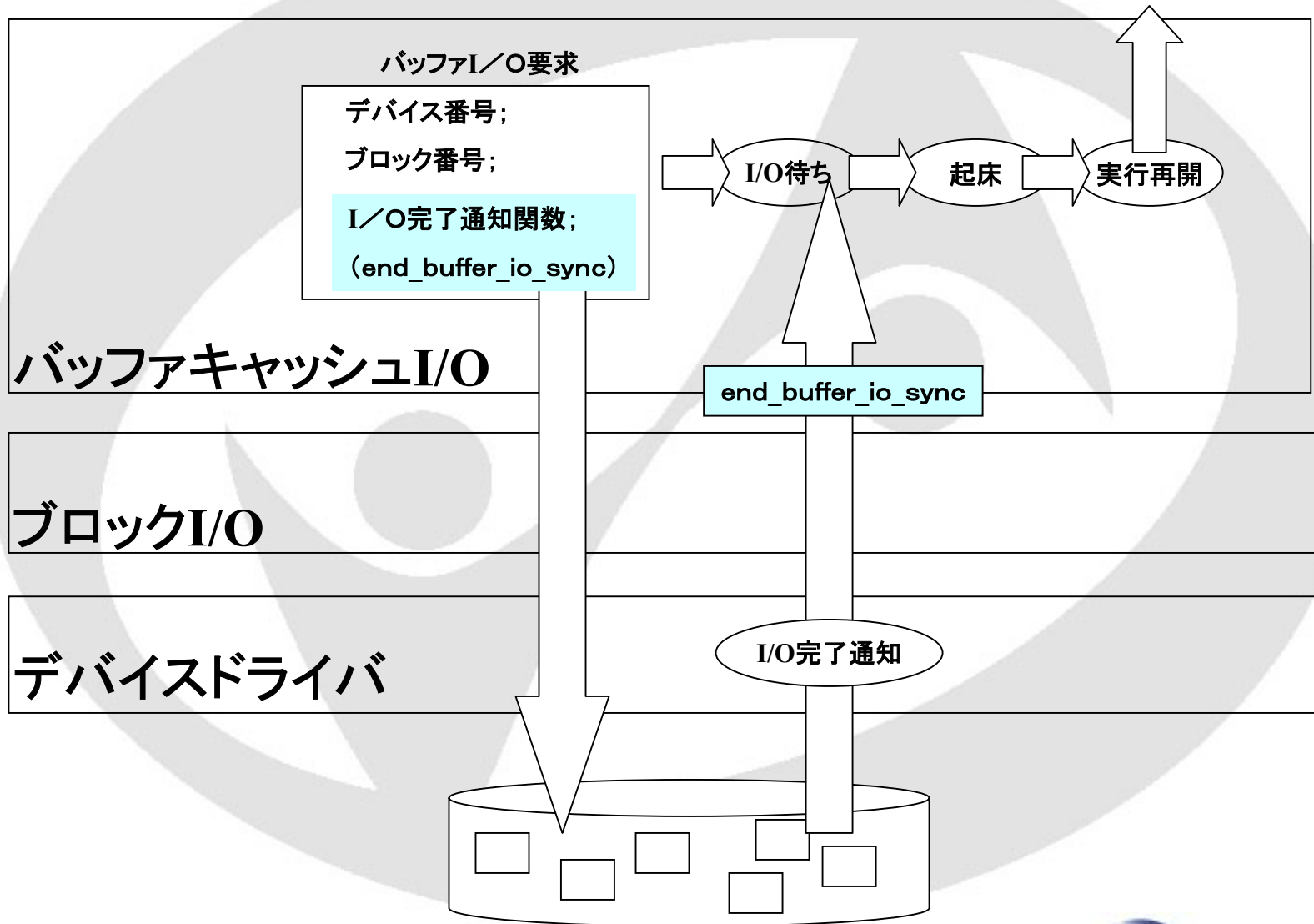


ページキャッシュ
と
バッファキャッシュ

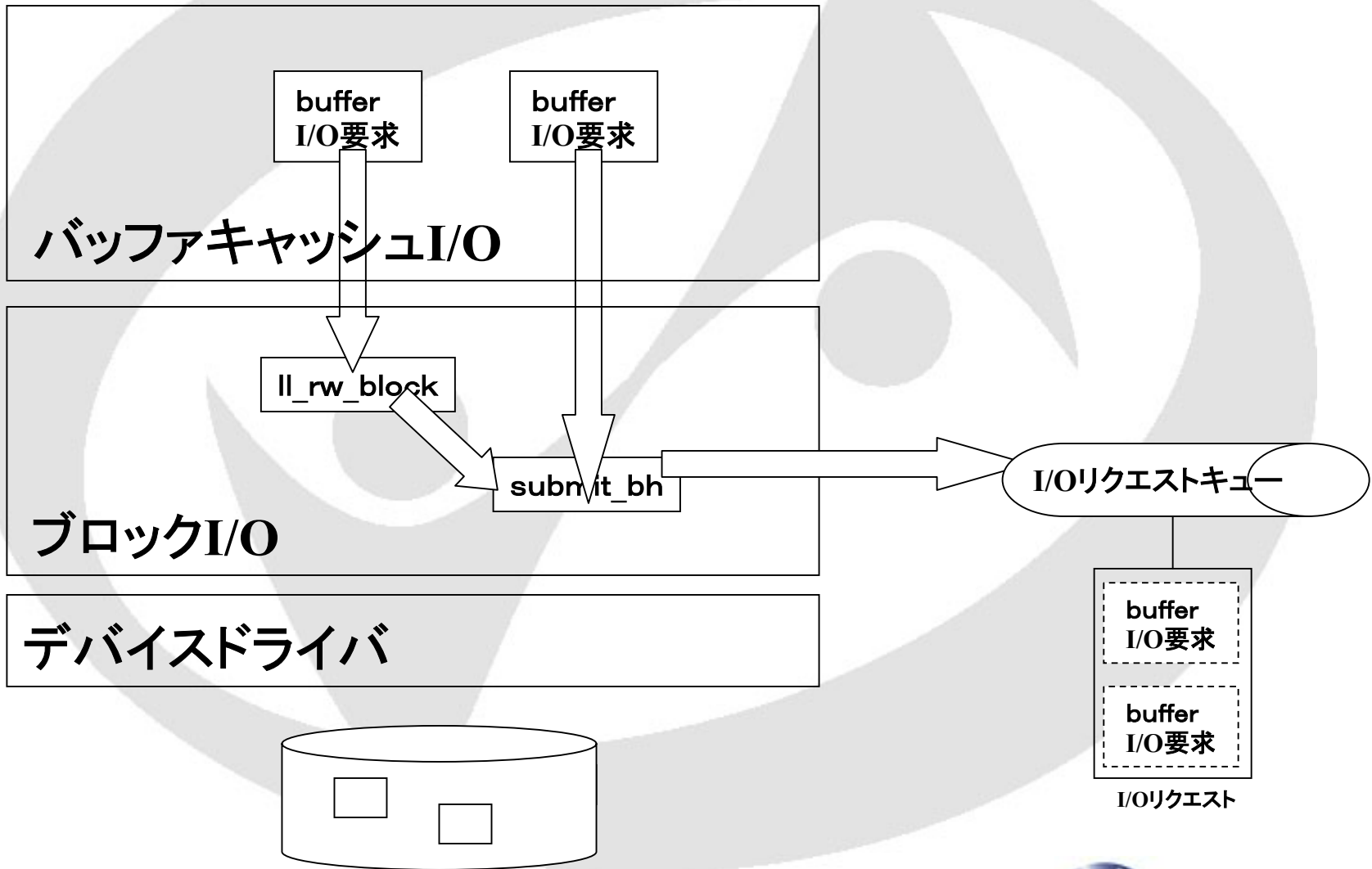
ファイルI/Oの処理階層



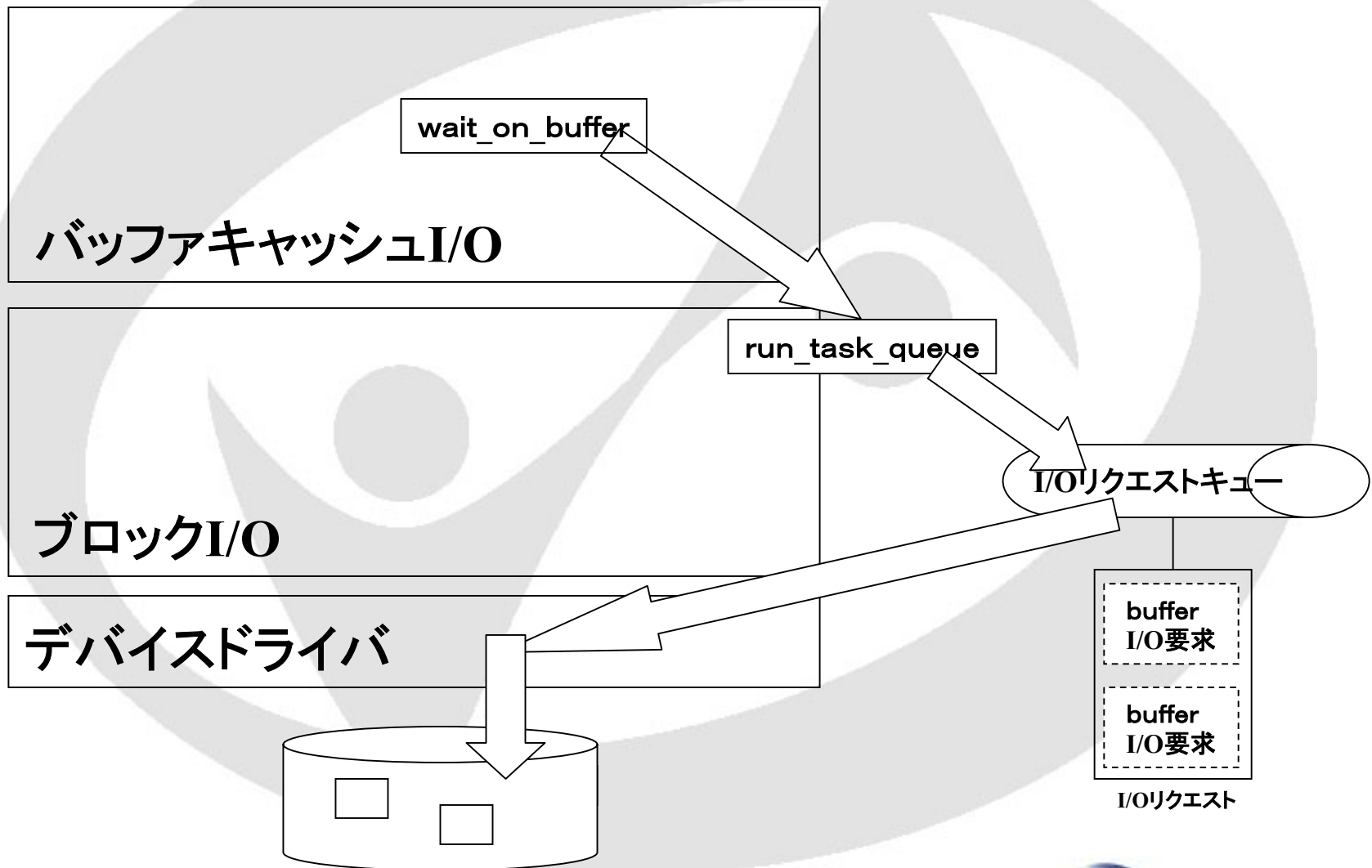
バッファキャッシュI/Oの概要



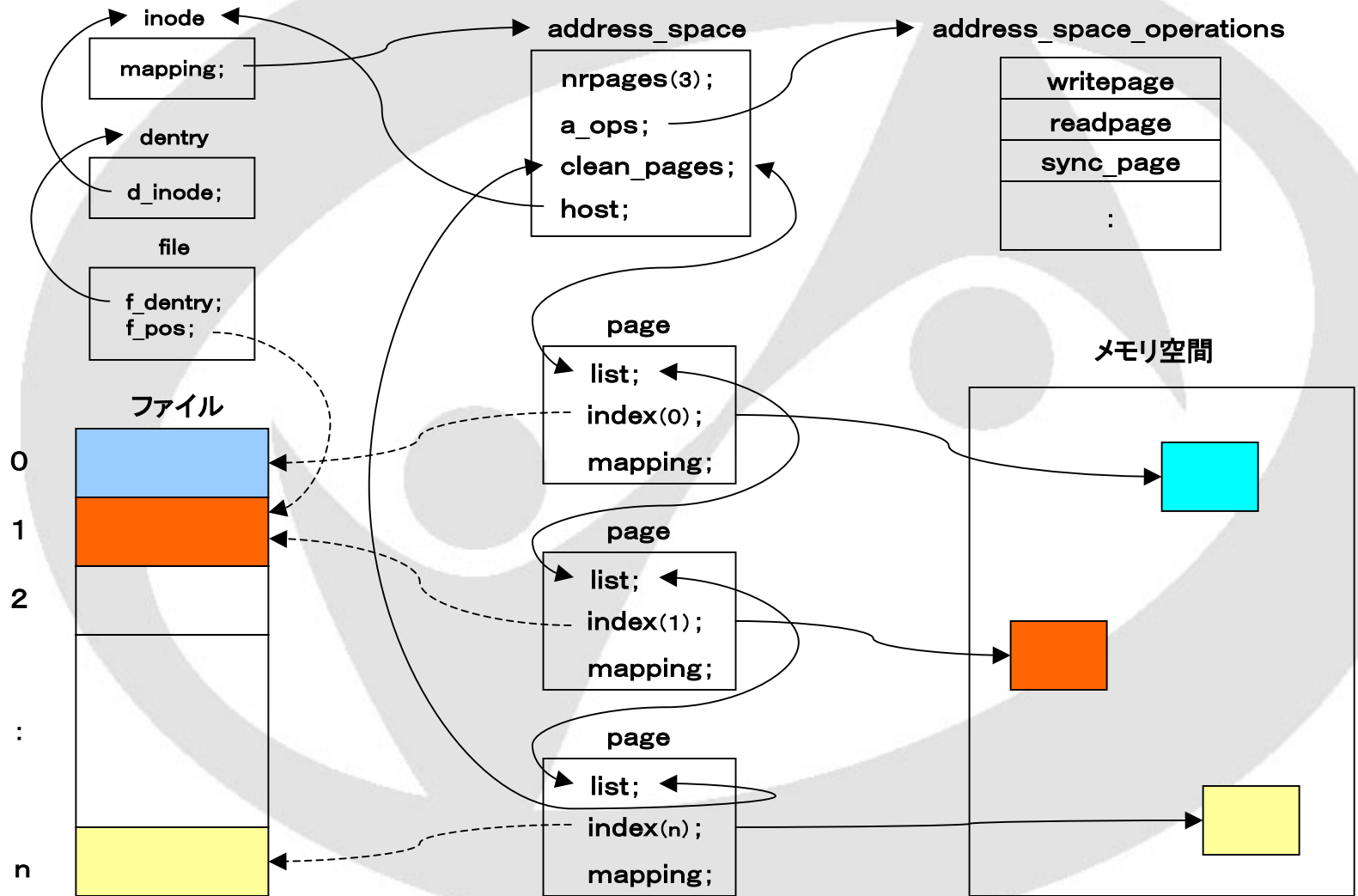
バッファのI/O要求



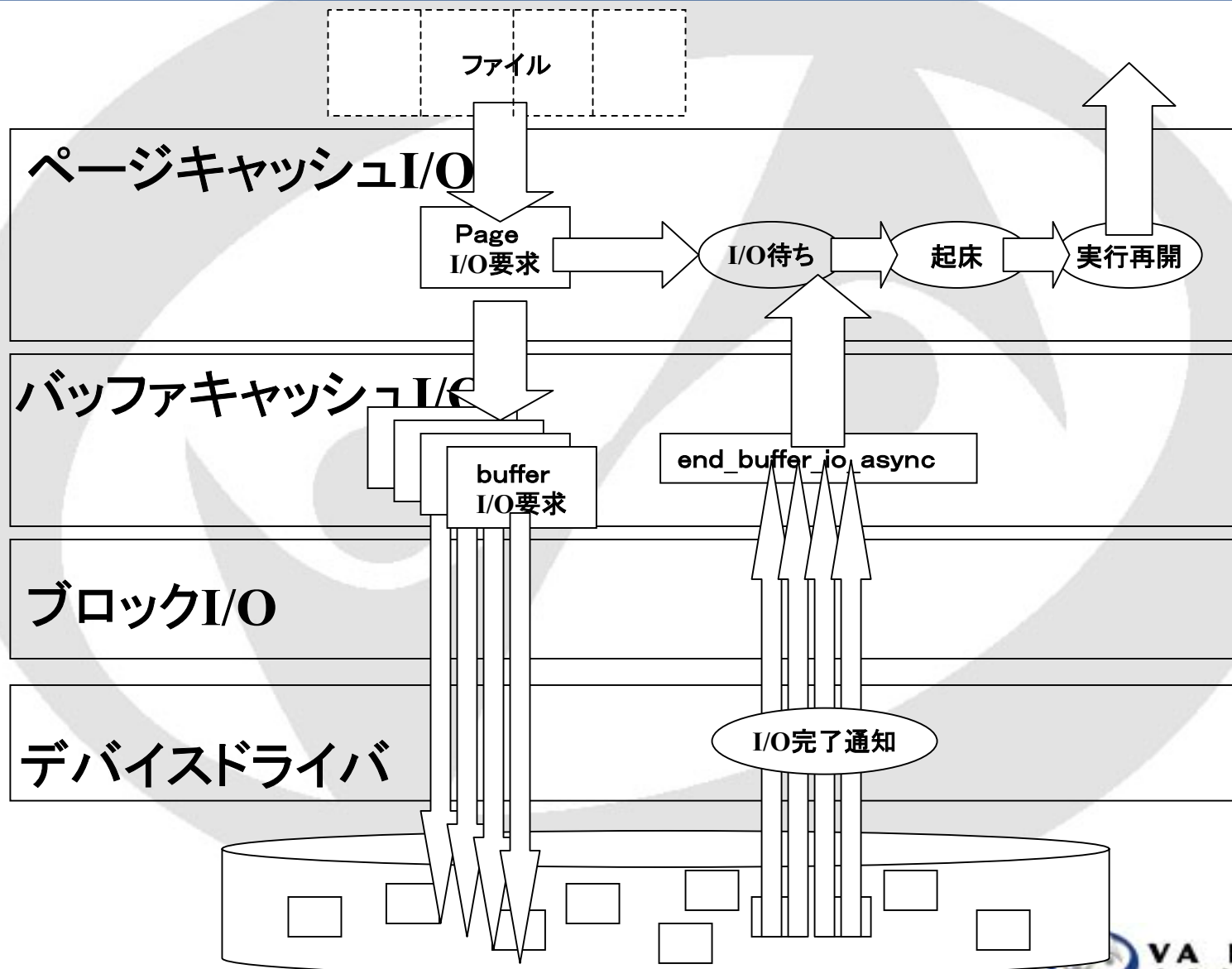
バッファのI/O実行



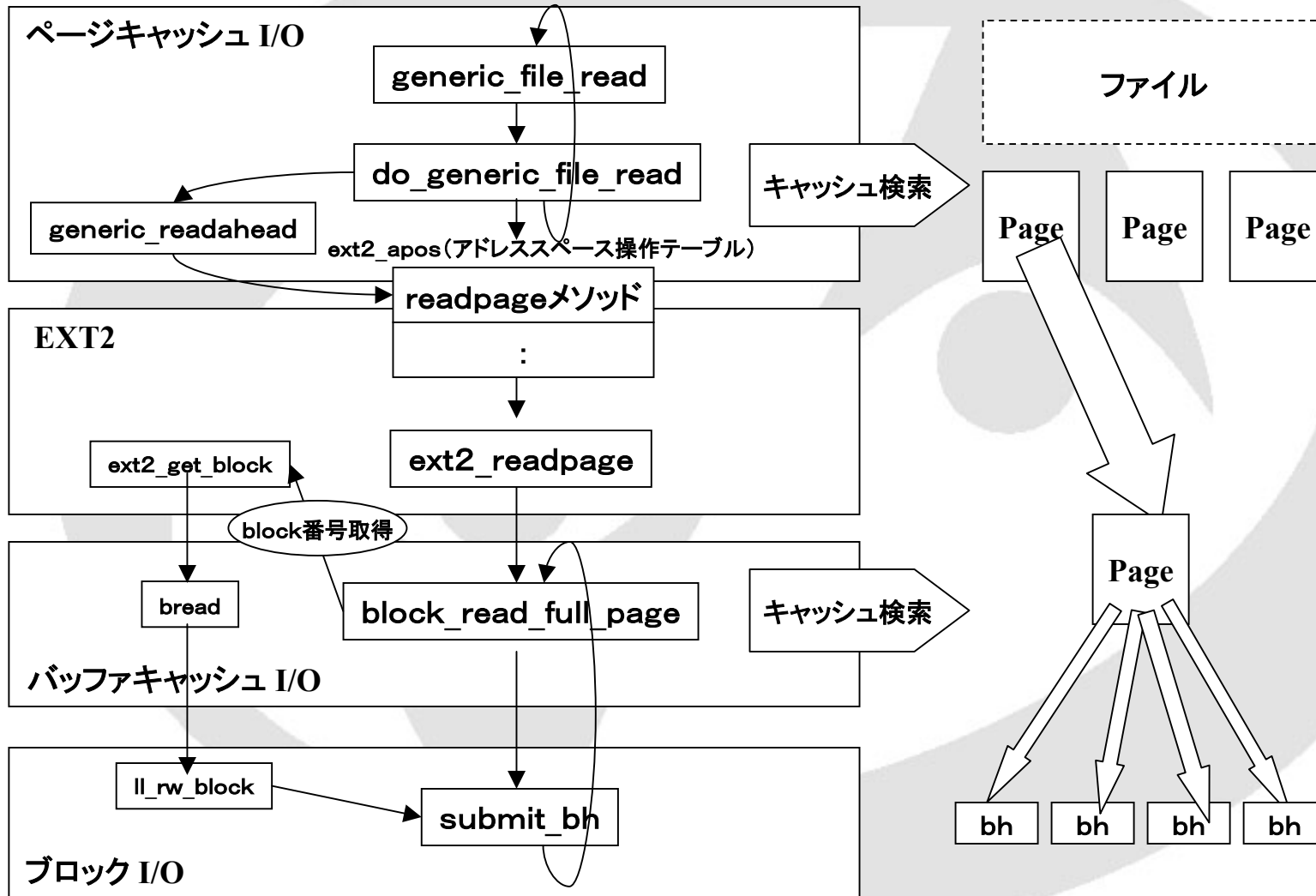
ファイルとページキャッシュ



ページキャッシュI/Oの概要



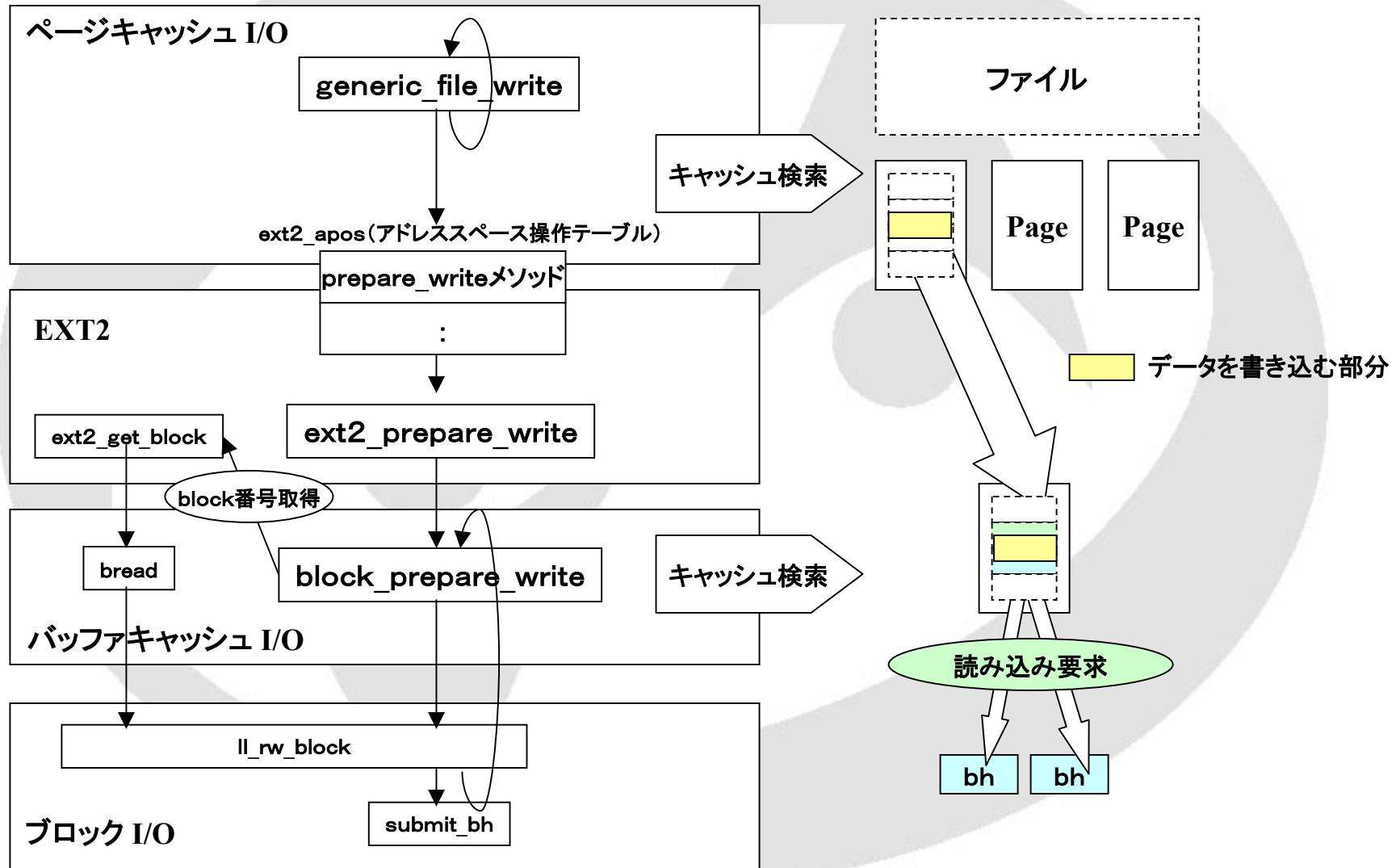
通常ファイル読み込みの流れ



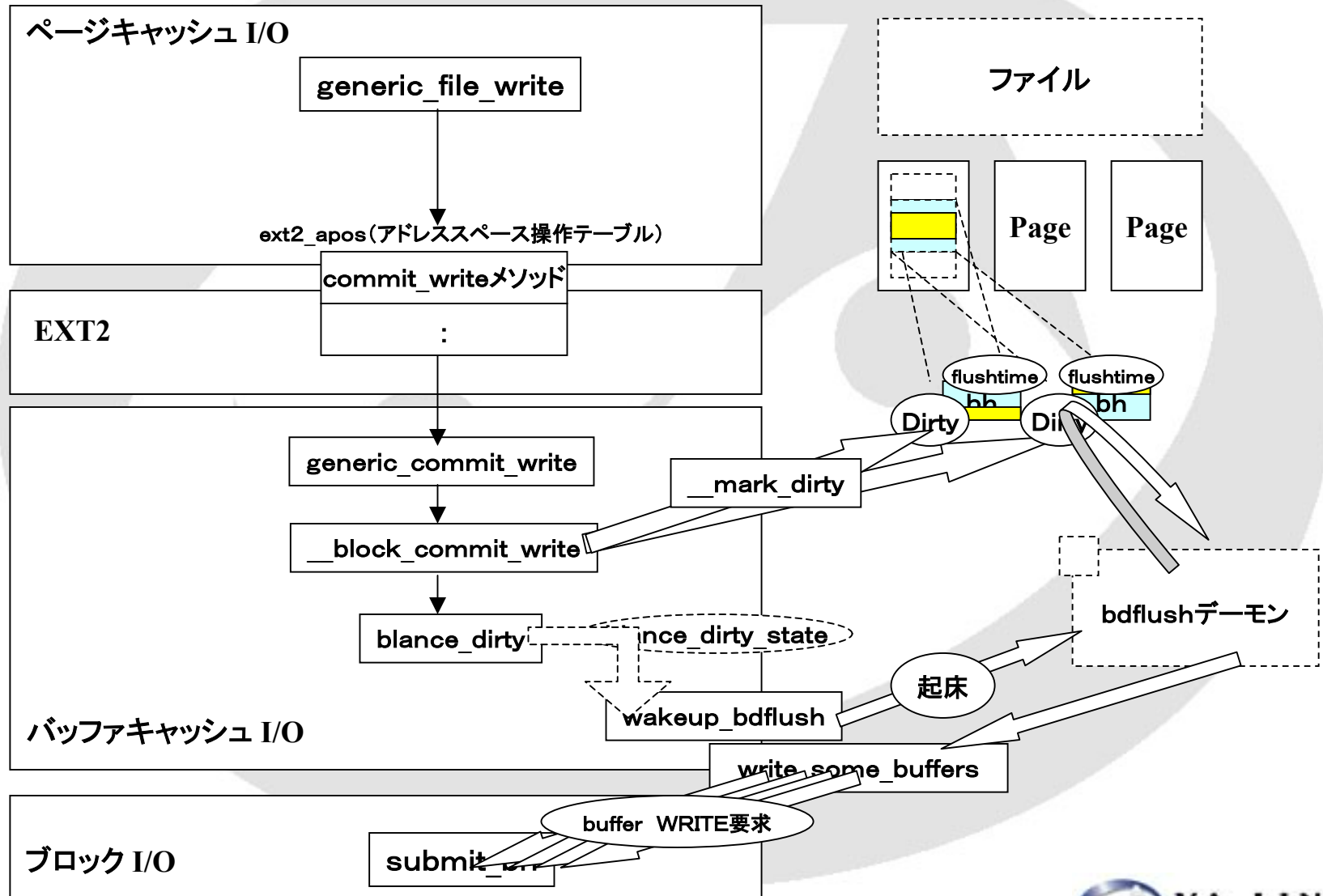
通常ファイルの書き込み

- `prepare_write`
- 書き込みデータコピー
- `commit_write`

通常ファイル書き込みの流れ (prepare_write)



通常ファイル書き込みの流れ(commit_write)



bdflushとkupdateデーモン

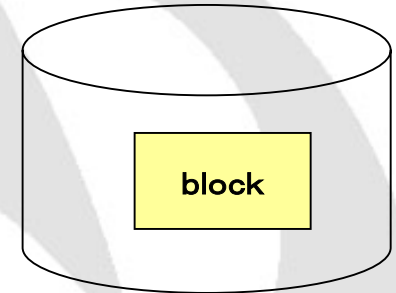
- **bdflushデーモン**
 - ー Dirty属性(遅延書き込み)のバッファが一定数以上になったらディスクに書き戻す
- **kupdateデーモン**
 - ー Dirty属性になって一定時間を経過したバッファをディスクに書き戻す
- **その他のディスクへの書き戻し処理**
 - ー updateコマンド(bdflushシステムコール)
 - ー syncコマンド(syncシステムコール)
 - ー 空きメモリの枯渇

ブロックI/O共通レイヤ

バッファI/OとブロックI/O

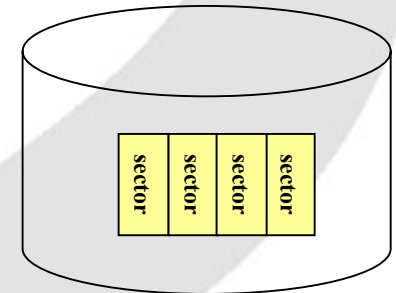
バッファI/O

- ブロック単位で処理
- ブロックサイズはファイルシステム依存
(ページサイズは超えられない)

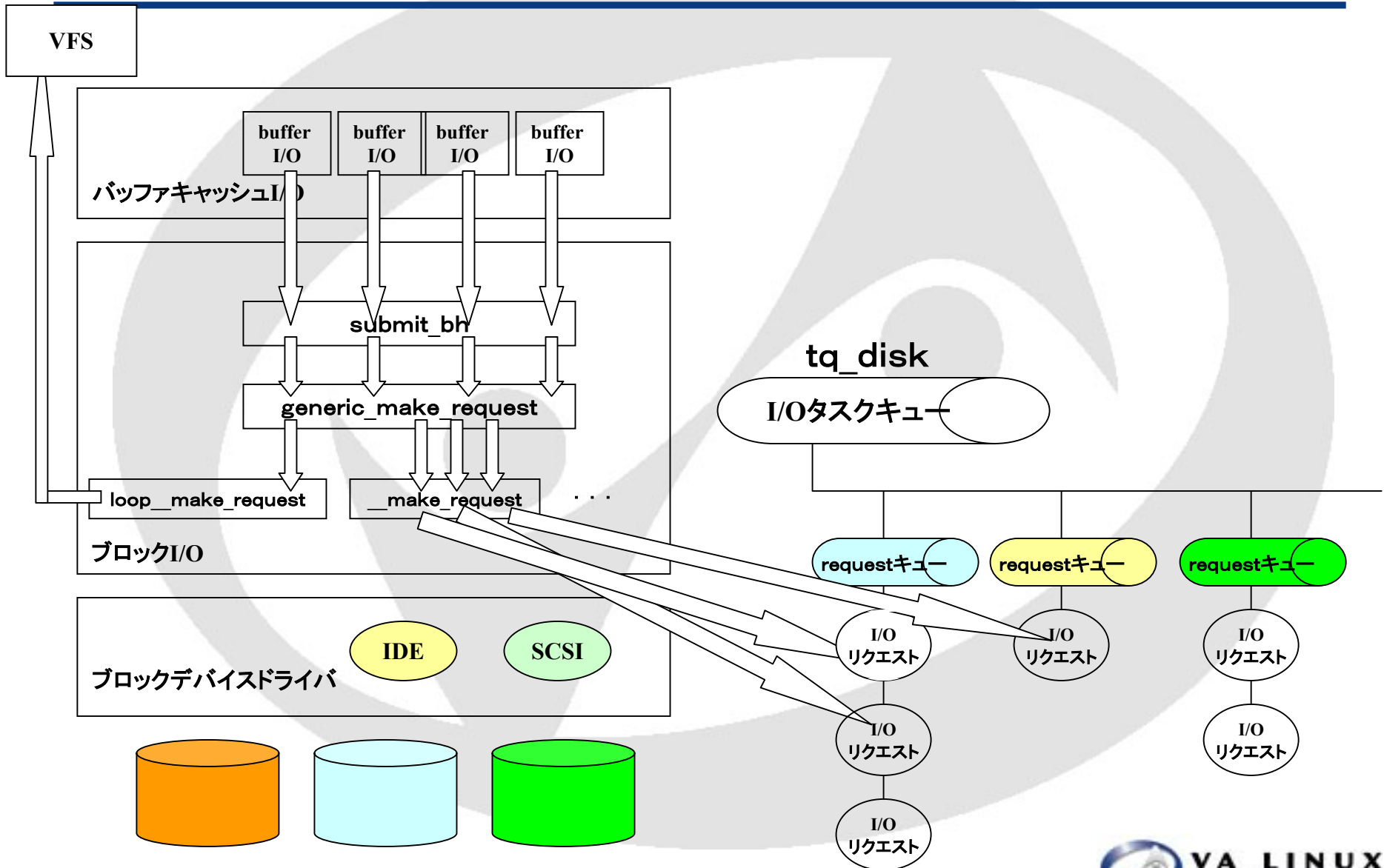


ブロックI/O共通レイヤ

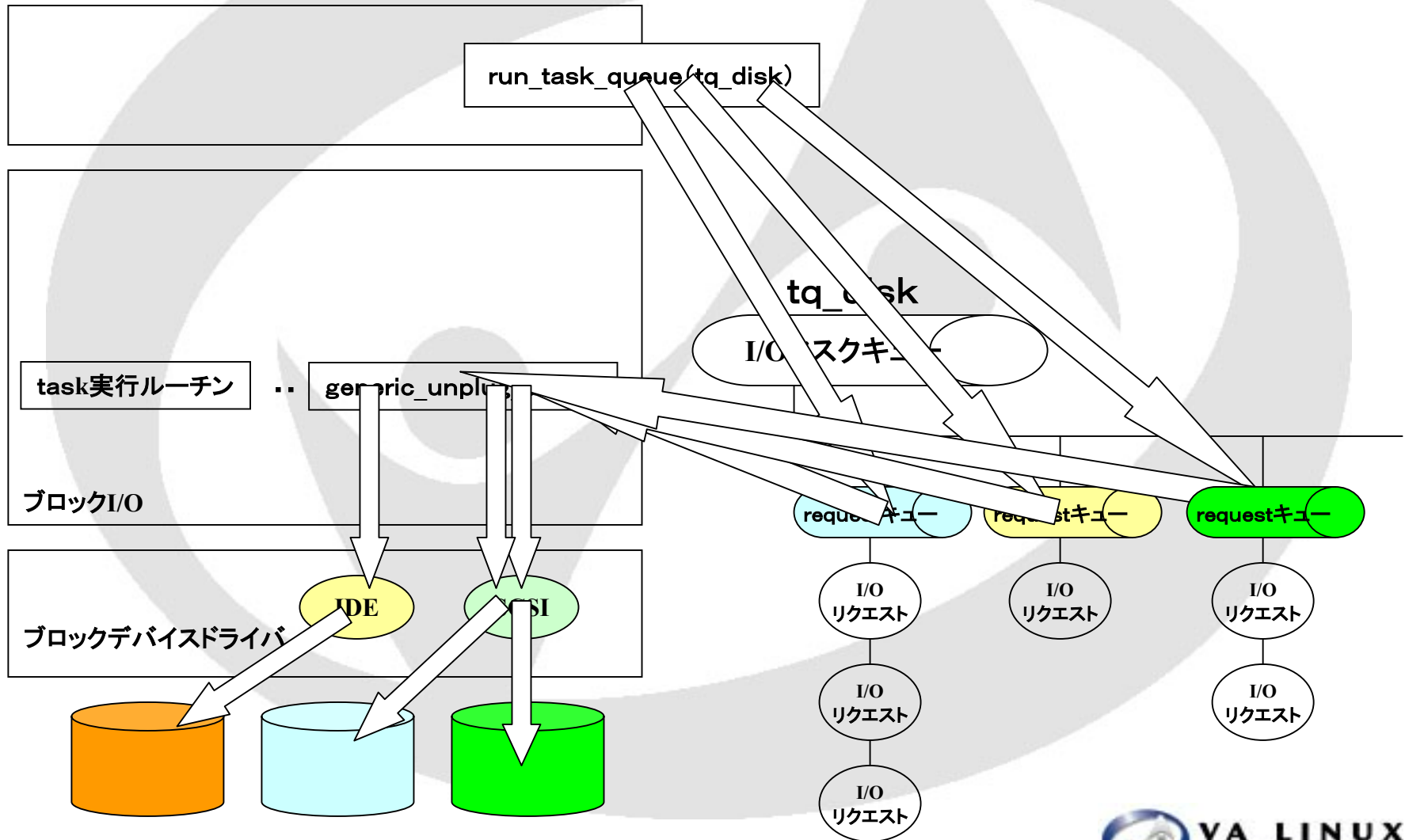
- セクタ単位で処理
- セクタサイズは512バイト
(ファイルシステムサイズの制限)



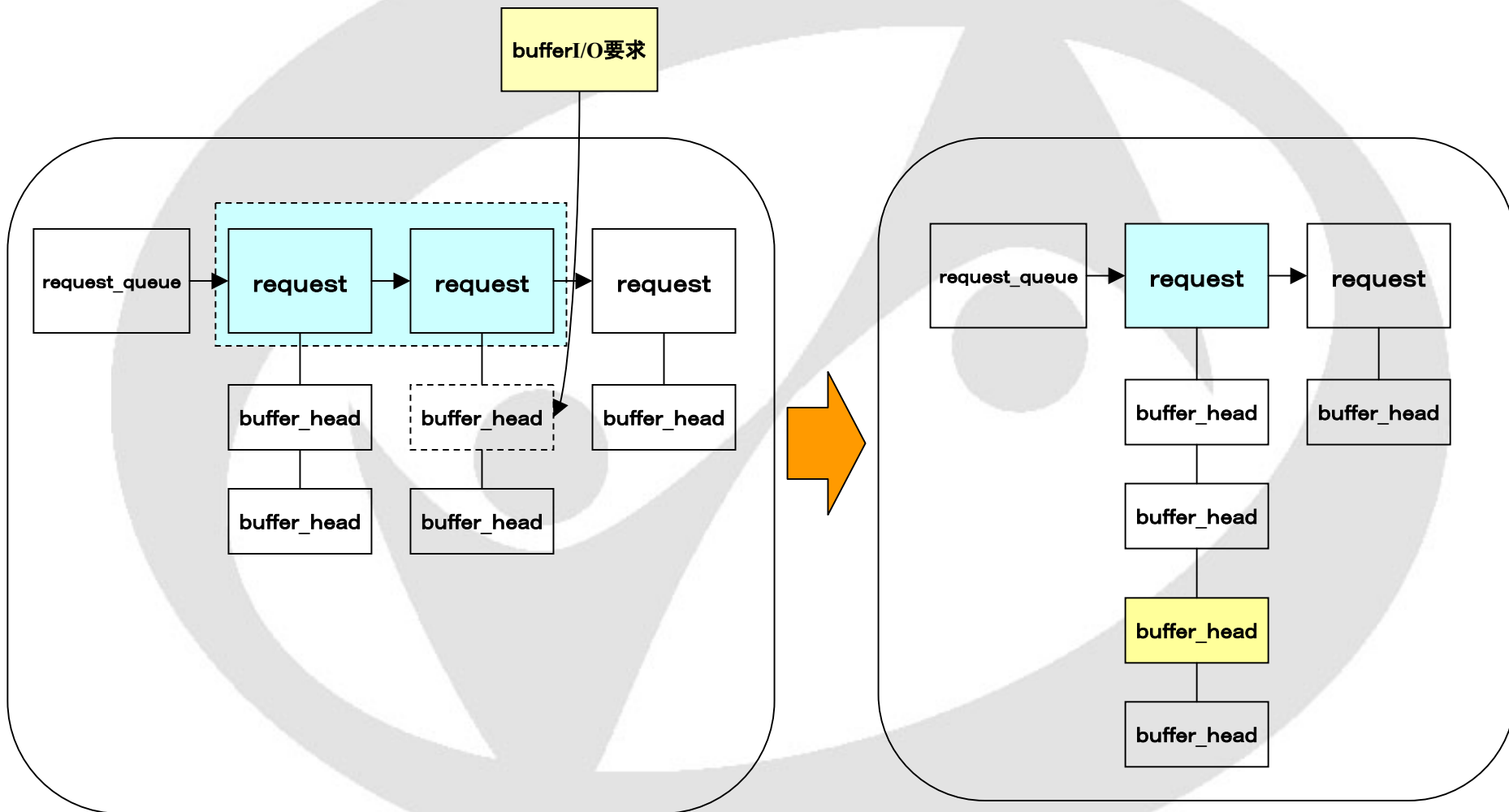
I/Oリクエストの作成



I/Oタスクの実行



ブロックI/Oのクラスタリング





おわり

<http://valinux.co.jp/>